😎 파일경로 세팅 및 파일 불러오기

In [19]:

```
import os

os.chdir('/' + os.path.join('Users','sindohyeon','Desktop','Rtest'))
print(os.getcwd())
```

```
/Users/sindohyeon/Desktop/Rtest
```

In [20]:

```
file = open('openAPI.txt', 'r')
# file 객체에서 문자열을 읽은 후 변수 x 에 저장
x = file.read()
# 변수 x 출력 print(x)
```

In [21]:

```
from bs4 import BeautifulSoup
import re
import nltk
```

In [22]:

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

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/sindohyeon/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[22]:

```
True
```

In [23]:

```
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
ps = PorterStemmer()
```

In [24]:

```
review_text = BeautifulSoup(x,'html.parser').get_text() #HTML 태그 삭제
```

In [26]:

```
letters_only = re.sub('[^a-zA-Z]',' ',review_text) #영문자가 아닌 문자는 공백 변환
```

In [27]:

```
words = letters_only.lower().split() #소문자 변환
```

In [28]:

```
stops = set(stopwords.words('english')) #Set 으로 변환
```

In [29]:

```
meaningful_words = [w for w in words if not w in stops] #Stopwords 제거
```

In [30]:

```
stemming_words = [ps.stem(w) for w in meaningful_words] #어간추출
```

```
len(stemming_words)
set1=set(stemming_words)
out = ' '.join(stemming_words)
```

😎 1 번 문제

In [32]:

#1-1. 전처리 작업 5 단계에서 얻은 단어 가운데 중복을 제거하면 총 몇개의 단어가 나오는가?

In [33]:

```
len(list(set(stemming_words)))
```

Out[33]:

답: 487

In [34]:

#1-2. api 라는 단어는 총 몇회 나오는가?

In [35]:

```
stemming_words.count('api')
```

Out[35]:

답: 77

In [36]:

#1-3. 5 번째로 빈도가 많은 단어는 무엇인가?

In [37]:

```
from collections import Counter
print(Counter(stemming_words).most_common(5)[4]) #답: develop
```

('develop', 22)

In [38]:

#1-4. 앞에서 구한 단어-빈도 관계와 아래의 코드를 이용하여 빈도 출현 횟수 상위 20 개 단어 빈도 그래프를 그리시오.

In [39]:

```
most_common_words = Counter(stemming_words).most_common(20)
```

In [40]:

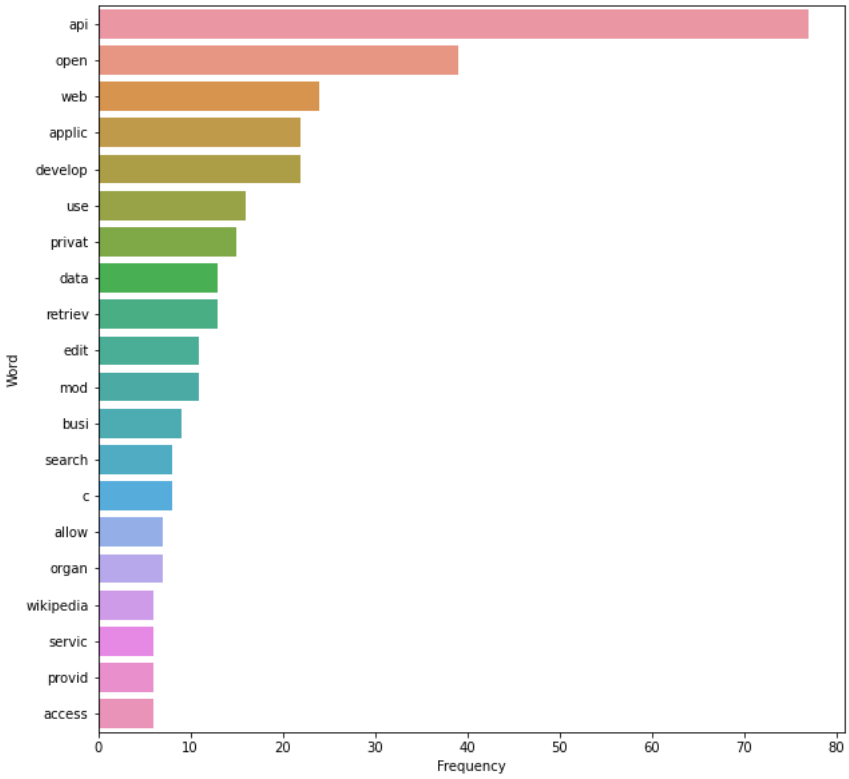
```
import matplotlib.pyplot as plt
import seaborn as sns
x, y= [], []
for word,count in most_common_words[:20]:
  x.append(word)
```

```
  y.append(count)
print(x,y)
plt.figure(figsize=(10,10)) #가로 세로 9 인치로 설정
ax = sns.barplot(x=y,y=x)
ax.set(xlabel = 'Frequency', ylabel = 'Word') #레이블 설정
```

```
['api', 'open', 'web', 'applic', 'develop', 'use', 'privat', 'data', 'retriev', 'edit', 'mod', 'busi', 'search', 'c', 'allow', 'organ', 'wikipedia', 'servic', 'provid', 'access'] [77, 39, 24, 22, 22, 16, 15, 13, 13, 11, 11, 9, 8, 8, 7, 7, 6, 6, 6, 6]
```

Out[40]:

```
[Text(0.5, 0, 'Frequency'), Text(0, 0.5, 'Word')]
```



In [41]:

#1-5 아래의 함수를 이용하여 2-gram 을 구한 뒤, {open api}라는 단어 조합의 빈도를 구하시오.

In [42]:

```
from nltk.tokenize import word_tokenize
from nltk.util import ngrams
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]       /Users/sindohyeon/nltk_data...
[nltk_data]     Package punkt is already up-to-date!
```

Out[42]:

```
True
```

In [43]:

```
def get_ngrams(text, n):
  n_grams = ngrams(word_tokenize(text), n)
  return [ ' '.join(grams) for grams in n_grams]

two_gram = get_ngrams(out,2)

from collections import Counter
count2=Counter(two_gram)
print(count2["open api"]) #open api 가 몇번 출력되는지 확인
```

답: 30

In [44]:

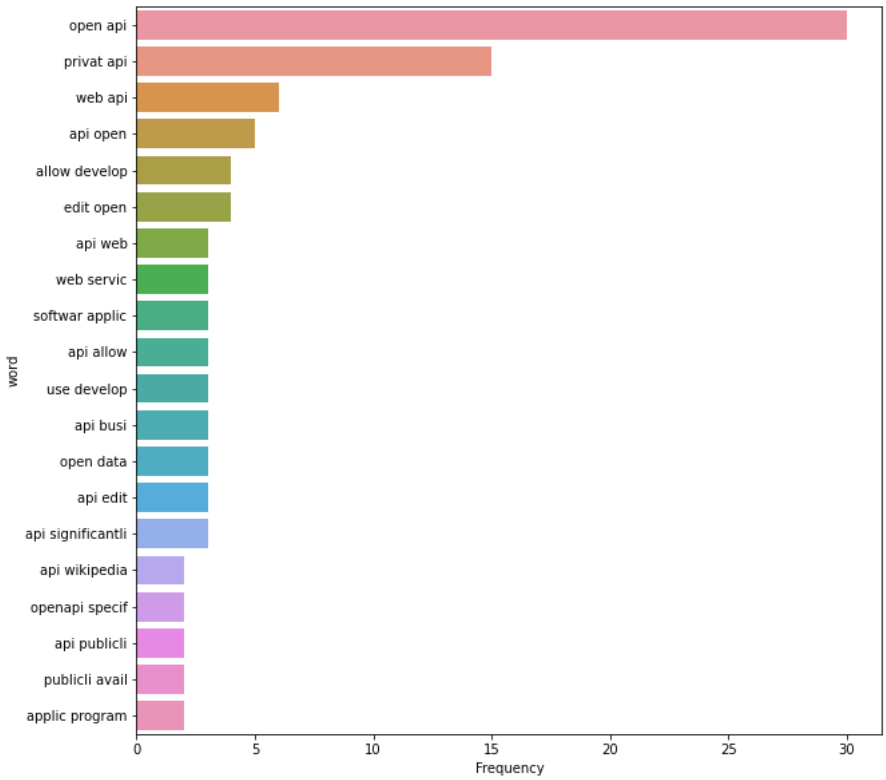
#1-6 위의 (5)에서 구한 2 단어-빈도 관계를 이용하여 (4)와 같은 빈도 그래프를 그리시오.

In [45]:

```
two_common_words = count2.most_common(20)
a,b=[],[]
for w, c in two_common_words[:20]:
      a.append(w)
      b.append(c)
plt.figure(figsize=(10,10))
ab = sns.barplot(x=b,y=a)
ab.set(xlabel='Frequency', ylabel='word')
```

Out[45]:

```
[Text(0.5, 0, 'Frequency'), Text(0, 0.5, 'word')]
```



😎 2 번 문제

---

[59]

3초

```
!pip install konlpy

import pandas as pd
import matplotlib.pyplot as plt
import urllib.request
```

```
Requirement already satisfied: konlpy in /usr/local/lib/python3.7/dist-packages (0.6.0)
Requirement already satisfied: JPype1>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from konlpy) (1.3.0)
Requirement already satisfied: lxml>=4.1.0 in /usr/local/lib/python3.7/dist-packages (from konlpy) (4.2.6)
Requirement already satisfied: numpy>=1.6 in /usr/local/lib/python3.7/dist-packages (from konlpy) (1.21.6)
```

Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from JPype1>=0.7.0->konlpy) (4.2.0)

---

[60]

5초

```
!pip install --upgrade gensim
from gensim.models.word2vec import Word2Vec
!conda install -c conda-forge jpype1 --yes
!pip install konlpy --yes
from konlpy.tag import Okt
```

```
Requirement already satisfied: gensim in /usr/local/lib/python3.7/dist-packages (4.2.0)
Requirement already satisfied: smart-open>=1.8.1 in /usr/local/lib/python3.7/dist-packages (from gensim) (6.0.0)
Requirement already satisfied: numpy>=1.17.0 in /usr/local/lib/python3.7/dist-packages (from gensim) (1.21.6)
Requirement already satisfied: scipy>=0.18.1 in /usr/local/lib/python3.7/dist-packages (from gensim) (1.4.1)
/bin/bash: conda: command not found

Usage:
  pip3 install [options] <requirement specifier> [package-index-options] ...
  pip3 install [options] -r <requirements file> [package-index-options] ...
  pip3 install [options] [-e] <vcs project url> ...
  pip3 install [options] [-e] <local project path> ...
  pip3 install [options] <archive url/path> ...

no such option: --yes
```

---

[89]

0초

```
okt = Okt()
file_name = '코로나_naver_news1.csv'
news = pd.read_csv(file_name, engine="python")

import re
news['description'] = news['description'].apply(lambda x: re.sub(r'[^ ㄱ-ㅎ| 가-
  ]+', " ", x))
news.head()
```

	Unnamed: 0	title	description	title_label	description_label
0	0	결국 증세론 먼저 꺼내든 증세없는 기본소득 불가능	코로나 발 경제 위기 대응을 위해 돈 쓸 곳은 늘어났지만 국세 수입은 줄어들면서...	0	0
1	1	창녕군 창녕형 비대면 선별진료소 운영	지난 일 창녕군보건소 앞에 설치한 선별진료소에서 검사자가 체온을 측정하고 있다...	0	0
2	2	모바일 메인 홍보 모델로 설현 선정	한편 설현 은 최근 코로나 바이러스를 다룬 시리즈 세계적 유행 에...	0	0
3	3	김병민 기본소득도 필요하면 논의 테이블에 올려야 인터뷰	변화의 핵심 중에서는 우리 사회가 신종 코로나 바이러스 감염증 코로나 의 위기를 ...	0	1
4	4	이재갑 장관 고용안정지원금 서울센터 방문	이재갑 고용노동부 장관은 월 일 수 시에 코로나 긴급 고용안정지원금 서울 센터 서...	0	0

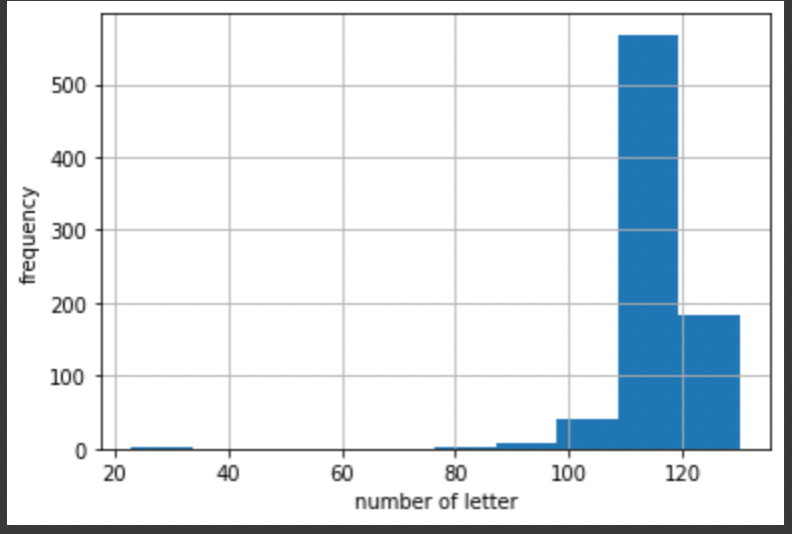
2-1 각 description 내용의 글자 수에 대한 빈도 (히스토그램)를 아래 코드를 이용해서 그리시오.

[90]

0초

```
ax = news['description'].str.len().hist()
ax.set_xlabel('number of letter')
ax.set_ylabel('frequency')
```

```
Text(0, 0.5, 'frequency')
```



2-2. description 내용 전체를 하나의 문자열로 만들고, KoNLPy 의 Okt(Open Korea text)를 이용하여 [명사(nouns)]만 추출한 뒤 가장 빈도가 많은 상위 4 번째 단어와 해당 빈도를 구하시오.

---

[91]

4초

```
text = ''.join(news['description'])
getNouns = okt.nouns(text)


from collections import Counter
wd_noun1=Counter(getNouns)
most_common_words0 = wd_noun1.most_common(4)
print(most_common_words0[3])
```

답: ('바이러스', 217)

2-3. 위의 (2)에서 얻은 명사 가운데 가장 빈도가 많은 2 글자 이상 단어 상위 4 개와 해당 빈도들을 구하시오.

---

[92]

0초

```
word_list = []
for w in getNouns:
  if len(w) > 1:
    word_list.append(w)

from collections import Counter
wd_noun1=Counter(word_list)
most_common_words1 = wd_noun1.most_common(4)
print(most_common_words1)
```

답: [('코로나', 1322), ('바이러스', 217), ('신종', 197), ('감염증', 190)]

2-4. 위의 1 번 (5)의 2-gram 함수를 이용해서 상위 4 개 토큰과 해당 빈도들을 구하시오. 여기서 2- gram 가운데 하나라도 1 글자인 경우는 제외한다.

---

[93]

0초

```
 from nltk.tokenize import word_tokenize
 from nltk.util import ngrams
 import nltk
 nltk.download('punkt')
 from collections import Counter
```
```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

---

[97]

0초

```
 def get_ngrams(text, n):
   n_grams = ngrams(word_tokenize(text), n)
   return [ ' '.join(grams) for grams in n_grams]

 out = " ".join(word_list)
 two_gram = get_ngrams(out,2)
 count = Counter(two_gram)
 two_common_words = count.most_common(4)
 print(two_common_words)
```
답: [('코로나 바이러스', 211), ('신종 코로나', 195), ('바이러스 감염증', 187), ('감염증 코로나', 173)]

😎 3 번 문제

3-1. coin.zip 파일 안의 전체 txt 파일을 결합하여 coinzip.txt 라는 파일을 생성하는 코드를 아래의 프로그램을 이용하여 작성하고, 생성된 파일(coinzip.txt)을 제출하시오.

---

[98]

0초

```
 path= '/content/' # coinzip.txt 가 저장된 경로
 file_list = os.listdir(path)
 file_list_py = [file for file in file_list if file.endswith('.txt')]
 len(file_list_py)
 print(file_list_py)  # 파일의 이름만 불러옴
```
```
['openAPI.txt']
```

---

[99]
```
# (1)
with open('coinzip.txt', 'wb') as outfile:
    for f in file_list_py:
        with open(f, 'rb') as infile:
            outfile.write(infile.read())
```

---

[110]

0초

```
# (2)
with open('coinzip.txt', 'rt', encoding='utf-8') as result:
  data1 = result.read()

from bs4 import BeautifulSoup
import re
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
ps = PorterStemmer()

# 전처리 작업
letters_only = re.sub('[^a-zA-Z]', ' ', data1)
words = letters_only.lower().split()
stops = set(stopwords.words('english'))
meaningful_words = [w for w in words if not w in stops]
stemming_words = [ps.stem(w) for w in meaningful_words]
len(stemming_words)
set1=set(stemming_words)
out = " ".join(stemming_words) # 이 자료를 이용하여 워드클라우드 그리기

from collections import Counter
count=Counter(stemming_words)
type(count)
count.most_common(1) #가장 많은 단어: li, 빈도 352
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

답: [('li', 352)]

---

[106]

9초

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

def plot_cloud(wordcloud):
  plt.figure(figsize=(50,50))
  plt.imshow(wordcloud)
  plt.axis("off")

wordcloud = WordCloud(width=1000, height=1000,
                      random_state=1,
                      collocations=False,
                      stopwords = STOPWORDS).generate(out)

plot_cloud(wordcloud)
```