

In [77]:

```
import requests
from bs4 import BeautifulSoup as bsp
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm #그래프 폰트설정
import re # 정규식 표현을 위한 모듈
import random
import pytagcloud
import webbrowser
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from collections import Counter
import pyautogui
```

In [95]:

```

def text_cleaning(text):#한글만 남길 함수
    hangul=re.compile('[^ㄱ-ㅣ가-힣]+')#[^ㄱ-ㅣ가-힣] 한글과 띄어쓰기의 정규식 패턴 ^ => [안에 범위
    result=hangul.sub('',text) #한글과 띄어쓰기를 제외한 모든 글자 패턴을 ''빈 문자로 치환
    return result

#네이버 뉴스 검색한 검색 페이지
text=pyautogui.prompt("검색어를 입력하세요")#사용자로부터 검색어를 입력받고
newsTitles=[]
for number in range(0,1000,10):#100페이지 까지 나온 기사 제목만 추출
    #입력받은 검색어로 검색
    response=requests.get(f"https://search.naver.com/search.naver?where=news&sm=tab_pge&query={text}")
    html=response.text
    page=bs4.BeautifulSoup(html,'html.parser')

    if page.select('.not_found02'):#검색결과가 없다면 if 실행
        print(page.select_one('.not_found02').text)
        break

    else:
        newsTitleTag=page.select('.news_area a.news_tit')
        for a in newsTitleTag:
            title=a.text
            newsTitles.append(title)

#검색결과가 없었다면 newsTitles에 추가된 title이 없으므로
if len(newsTitles)==0:
    pass
else:
    #뉴스페이지가 100페이지까지 없으면 마지막 기사가 반복되어 나옴
    my_set = set(newsTitles) #중복되는 title을 제거하기위해 집합set으로 변환
    newsTitles = list(my_set) #list로 변환
    columns=['title']
    df = pd.DataFrame(columns=columns)

    for title in newsTitles:
        title=title.replace(' ','')#title에서 ,제거
        series = pd.Series(title, index=df.columns)#Series형태로 바꿈
        df = df.append(series, ignore_index=True)#df에 추가

    df['title']=df['title'].apply(lambda x : text_cleaning(x))

#말뭉치 생성
title_corpus="".join(df['title'].tolist())
title_corpus

from konlpy.tag import Okt

#konlpy의 형태소 분석기로 명사 단위의 키워드를 추출합니다.
nouns_tagger=Okt()
nouns=nouns_tagger.nouns(title_corpus)

count=Counter(nouns)
count

stopwords=[text]#해당 검색어는 말뭉치에서 제거하기위해 stopwords에 저장
remove_char_counter = Counter({x : count[x] for x in count if x not in stopwords})
remove_char_counter = Counter({x : remove_char_counter[x] for x in remove_char_counter if len(x)

from PIL import Image

```

#지정된 이미지로 cloud생성

```
news_img=np.array(Image.open("news_img2.png"))
wc=WordCloud(font_path='malgun',background_color='white',max_words=2000,
             mask=news_img, max_font_size=40,random_state=42)
gen=wc.generate_from_frequencies(remove_char_counter)
from wordcloud import ImageColorGenerator#지정된 이미지의 색상 반영
imgc_colors=ImageColorGenerator(news_img)
plt.figure(figsize=(20,10))
plt.imshow(wc.recolor(color_func=imgc_colors),interpolation='bilinear')
plt.axis('off')
plt.savefig(f'word_{text}_img.png')#검색어로 파일명 생성
plt.show()
```



In []: