네이버 지도로 정제된 주소(지오코드용 주소)로 지오코딩

In [11]:

```
import os
import urllib.request
import urllib.error
import sys
import pandas as pd
```

지오코드용주소, 사전화된주소(네이버 지도 검색용), 원데이터 주소가 담긴 csv 파일 import

In [6]:

```
final_illigal_addr= pd.read_csv(os.getcwd()+'\\csv\\final_illigal_addr.csv')
```

지오코드용 주소의 NaN값은 네이버 지도 앱에서 찾을 수 없는 주소

In [8]:

```
final_illigal_addr.head()
```

Out[8]:

	address_for_geo	dicted_alter_addr_lst	dicted_ori_addr_lst
0	명장동 시실로215번길	명장동 시실로215번길	명장동 시실로215번길
1	부산광역시 남구 문현동	문현4동 장고개로	문현4동 장고개로부근(곡각,횡단보도)
2	부산광역시 영도구 절영로 21 지번	대교동2가 인제요양병원	대교동2가 인제요양병원
3	NaN	문현1동 아이원통상	문현1동 아이원통상부근
4	온천동 1485	온천동 1485	온천동 1485

지오코드 검색용 주소와 사전화된 주소를 코딩하기 편하게 리스트로 변경

In [33]:

```
lst_geo = list(final_illigal_addr["address_for_geo"])
lst_alter = list(final_illigal_addr["dicted_alter_addr_lst"])
```

네이버 지도 api id와 passwd

In [34]:

```
client_id = "client_id"
client_secret = "client_passwd"
```

반복문으로 지오코드용 주소를 정제하여 네이버지도api에 쿼리 요청, 받은 json 파일에서 필요한 데이터 추출

In [ ]:

```
lon = []
lat = []
raw_data = []
for i in range(len(lst_geo)):
    #지오코드 주소 중 nan값과  '~번길' 포함된 값의 재정제 필요.
    if str(lst_geo[i]) == "nan":
        encText = urllib.parse.quote(lst_alter[i])
    elif lst_geo[i].find("동 ")>0 and lst_geo[i][-2:]=="번길":
        tmp = lst_geo[i].split("동 ")
        encText = urllib.parse.quote(tmp[1])
```

```
        encText = urllib.parse.quote(tmp[1])
    elif lst_geo[i].find("읍 ")>0 and lst_geo[i][-2:]=="번길":
        tmp = lst_geo[i].split("읍 ")
        encText = urllib.parse.quote(tmp[1])
    elif lst_geo[i].find("가 ")>0 and lst_geo[i][-2:]=="번길":
        tmp = lst_geo[i].split("가 ")
        encText = urllib.parse.quote(tmp[1])
    else:
        encText = urllib.parse.quote(lst_geo[i])

    url = "https://openapi.naver.com/v1/map/geocode?query=" + encText # json 결과
    request = urllib.request.Request(url)
    request.add_header("X-Naver-Client-Id",client_id)
    request.add_header("X-Naver-Client-Secret",client_secret)
    try:
        response = urllib.request.urlopen(request)
        rescode = response.getcode()
        if(rescode==200):
            response_body = response.read()
            raw_data.append(response_body.decode('utf-8'))
            #print(response_body.decode('utf-8'))
            raw_data.append(response_body.decode('utf-8'))
            target_start_num = response_body.decode('utf-8').find('"부산광역시 ')
            #받아온 json 파일에 "부산광역시" 데이터면 위도/경도 추출
            if target_start_num >0:
                new_body = response_body.decode('utf-8')[target_start_num:]
                x_start_num = new_body.find('"x"')
                new_body = new_body[x_start_num:]
                x_end_num = new_body.find(',\n')
                x = new_body[5:x_end_num] #lon
                y_start_num = new_body.find('"y": ')
                new_body = new_body[y_start_num:]
                y_end_num = new_body.find("\n")
                y=new_body[5:y_end_num] #lat
                lon.append(x)
                lat.append(y)
            else:
                lon.append(0.0)
                lat.append(0.0)
        else:
            #print("Error Code:" + rescode)
            raw_data.append(rescode)
    #검색이 안될 경우 위도경도는 0
    except urllib.error.HTTPError:
        #print('can not find the url')
        #print('null')
        lon.append(0.0)
        lat.append(0.0)
```

정제된 위도 경도와 json파일을 csv형태로 export

In [112]:

```
final_illigal_addr["longitude"] = lon
final_illigal_addr["latitude"] = lat
pd.DataFrame.to_csv(final_illigal_addr,os.getcwd()+'\\csv\\final_illigal_addr_lonlat.csv',sep=",",
index=False)
```

In [113]:

```
geocoding_raw_data = pd.DataFrame({"raw_data":raw_data})
pd.DataFrame.to_csv(geocoding_raw_data,os.getcwd()+"\\csv\\geocoding_raw_data.csv",sep=",",index=Fa
lse)
```