# 융합SW 최신기술:
# PART 3 - Project1

경영학과 20122682
류성호

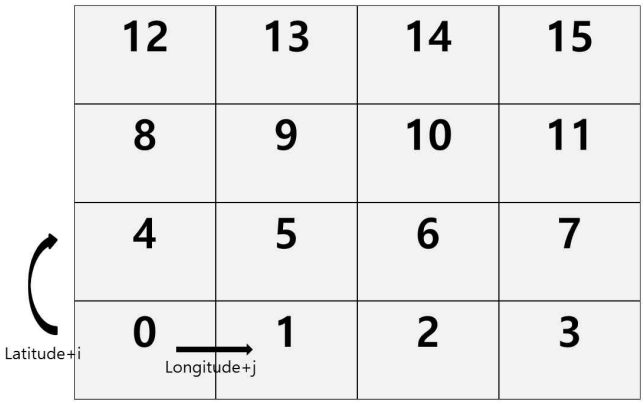
과제 :

1. tmap 개발자 센터 API를 이용하여 실시간 교통정보 가져오기

- tmap에서 제공하는 실시간 교통정보를 xml 파일의 형식으로 다운 받기 위해서는 우선 tmap에서 제공하는 APP_KEY를 받아야 한다.

```
TMAP_APP_KEY = "0a556b9e-906f-335c-9e4b-c3ad9091a103"
```

- 얻고자 하는 지도 정보(위도, 경도, zoomLevel)의 초기 값을 설정한다. 위도, 경도를 옮겨가면서, 우리가 지도를 통해 보는 한정된 영역에 대한 교통 정보를 4x4 매트릭스로 표현하기로 한다.



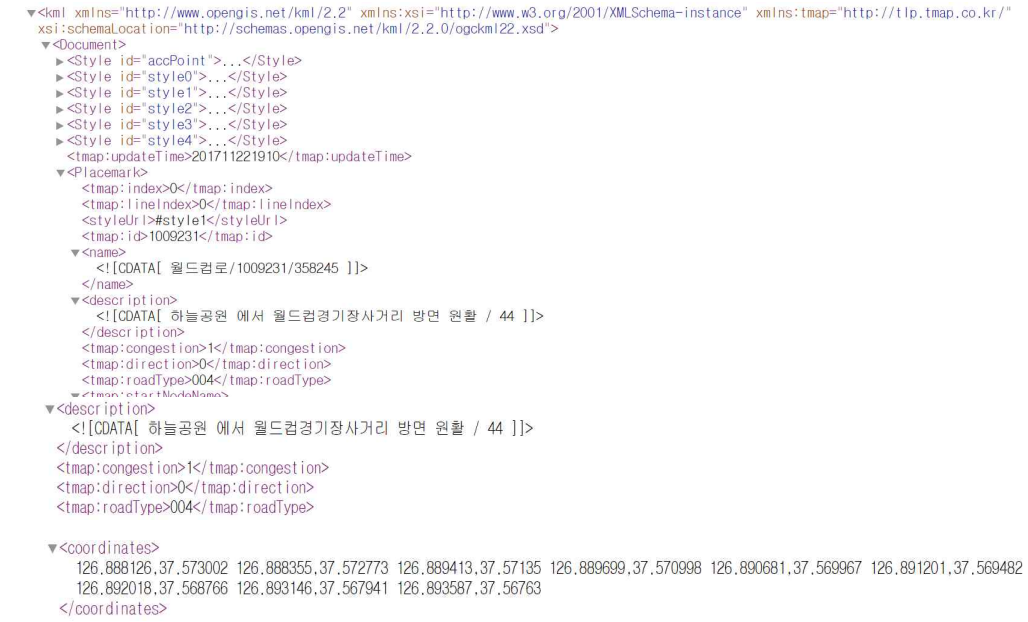

총 16칸의 볼 수 있는 영역을 설정하고, APP_KEY 값을 넣은 문자열로 url을 표현한 뒤, url.open 함수를 사용해 해당 url을 불러온다. 이중 for문을 사용해 칸과 열이 1씩 증가하는 수식을 표현했으며, 0.01을 곱해서 너무 급변하지 않게 증가폭을 조정한다. 그 후, KML 형식으로 제공되는 교통정보를 16개의 빈 xml 파일로 옮겨 저장한다.(traffic.xml)

```
for i in range(NUM):
    for j in range(NUM):
        AREA_INDEX  = NUM * i +j
        requestStr = "https://apis.skplanetx.com/tmap/traffic?version=1&minLat="+ ₩
            str(MIN_LAT+INT*i) + "&minLon=" + str(MIN_LON+INT * j)+ "&maxLat="+ str(MAX_LAT+INT *i)+ ₩
            "&maxLon="+ str(MAX_LON+INT * j)+"&zoomLevel="+ str(ZOOM_LEVEL)+"&appKey="+ TMAP_APP_KEY +₩
            "&reqCoordType=WGS84GEO&resCoordType=WGS84GEO&format=xml"
        in_file = ur.urlopen(requestStr)
        out_file = open(OUT_DIR+'traffic_%02d.xml' % (AREA_INDEX), 'wb')
        out_file.write(in_file.read())
        out_file.close()
```

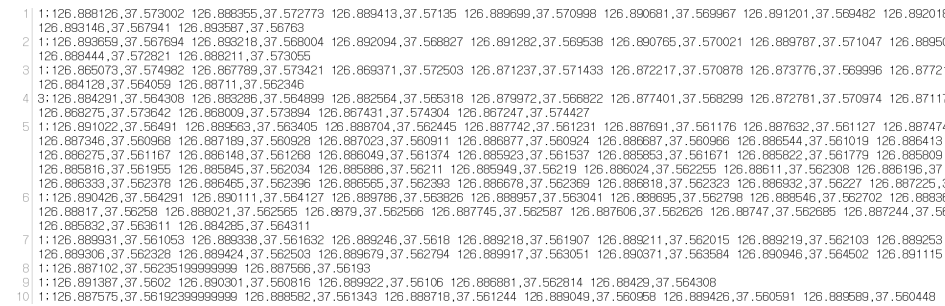
2. tmap 교통정보 xml파일에서 각 도로의 congestion level과 line string을 추출해내기

- 1번 과정에서 얻은 xml 파일들에는 <Placemark>란 tag가 있으며, 이 tag들에는 각 영역의 교통 정보들이 담겨있다.

- 그 중, 하위에 위치한 <congestion>, <coordinates> tag를 골라낸다. 이들은 각각 교통 혼잡도와 좌표를 나타내며,<congestion>은 1~5의 Level로, <coordinates>는 위도, 경도의 pairs로 표현된다.




- xml 파일에서 각 정보를 추출하기 위해서 BeautifulSoup 모듈을 사용하기로 한다. BeautifulSoup은 웹 페이지의 HTML 포맷 문서를 처리하기 쉽게 전처리 해주는 라이브러리로, xml 은 html과 유사한 점이 많으니 사용해도 무방하다.`soup = BeautifulSoup(f,'html5lib')`
- <congestion>, <coordinates> tag 들의 정보를 findAll() 함수를 사용해 리스트에 담고 이를 순서대로 congestion_coor_XX.txt 파일에 적었다. 일정한 형식을 가진 16개의 txt 파일이 만들어 진다.




3. 2번 과정에서 추출한 교통 정보를 Google map에 color-coded line으로 보여주기

- congestion_coor_XX.txt에 기록된 <congestion> <coordinates> 정보를 google map의 string 형식으로 바꿔줘야 한다. 우선 1번에서 tmap의 APP_KEY 받았듯이 google maps의 APP_KEY를 받는다.

```
GOOGLE_APP_KEY = 'AIzaSyAzgANCMQVVwbJW42KEb_NyhGtTHoSpFXs'
```

- 위 정보들을 가지고, string 형식의 url을 만들 수 있는데, 이 중 path라는 파라미터를 추가해 colored line을 그린다. line의 색은 congestion Level에 따라 5가지 색으로 표현하고, 각각 좌표는

하나의 <Placemark>에 해당하는 (위도, 경도) pair의 시작과 끝의 pairs만 추가한다. 즉, 하나의 xml 페이지에 기록된 여러 개의 Place mark 태그 안에 존재하는 congestion level과 coordinates를 모두 표현하되, 시작과 끝의 (위도, 경도) 만을 사용한다.

- color는 if-elif문을 사용해 구분했고, linestring로 구분된 좌표들의 집합 string은 변형하기 쉽게 리스트로 저장해 놓는다. 이때, 좌표는 위도와 경도가 거꾸로 되어 있기에, 다시 바꿔서 저장한다.

```
if congestion == '1':
    color.append('blue')
elif congestion == '2':
    color.append('green')
elif congestion == '3':
    color.append('yellow')
elif congestion == '4':
    color.append('orange')
elif congestion == '5':
    color.append('red')

mylist.append(linestring[:-1].split(" "))
```

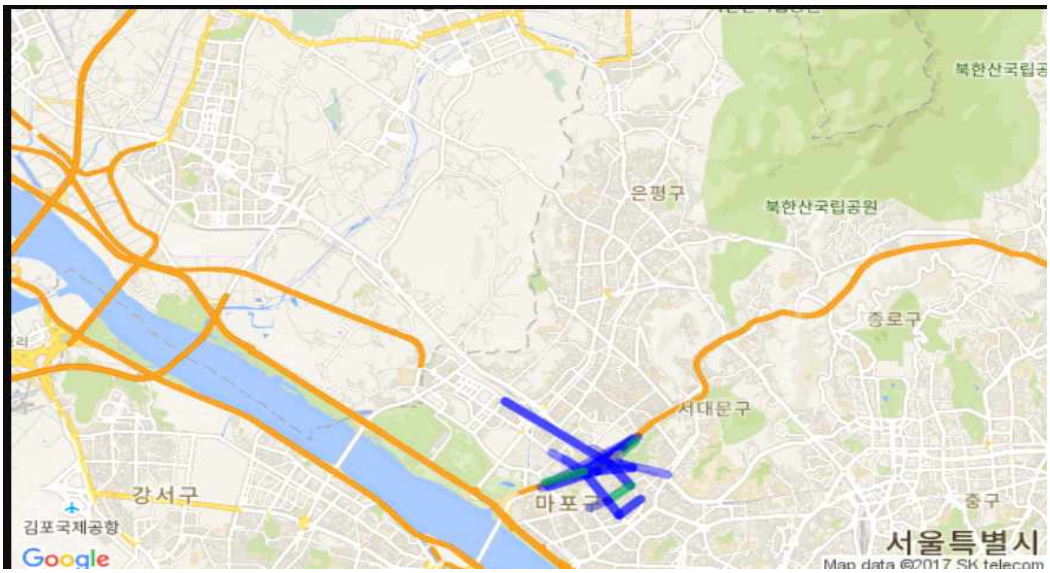
```
for x in range(0,len(mylist)):
    for y in range(len(mylist[x])):
        dd, ee = mylist[x][y].split(",")
        mylist[x][y] = (str(ee) + ',' + str(dd))
```

- 색과 (위도, 경도) pairs 로 구분된 정보들은 url 파라미터 중 path 속성에 string으로 추가되어 더해진다. 이 때, APP_KEY도 같이 추가해서 urlstr을 완성한다.

```
for p in range(len(color)):
    path += "&path=color:" +color[p] +"|weight:6|" +mylist[p][0] +'|' + mylist[p][len(mylist[p])-1]

urlstr = "https://maps.googleapis.com/maps/api/staticmap?center=" + CENTER+₩
    "&zoom=12&size=800x400&maptype=roadmap&sensor=false&key=" + GOOGLE_APP_KEY + path
```

- 이렇게 완성된 urlstr은 gmap_urlstr_XX.txt 에 저장하고, url을 web browser에서 실행하면 다음과 같은 화면이 나온다.

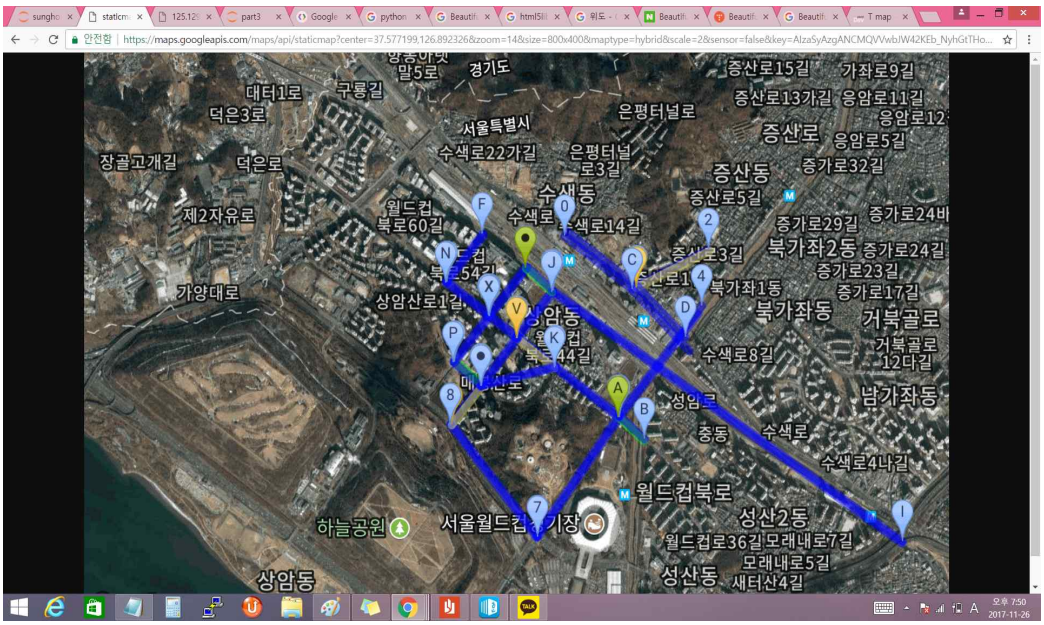


4. tmap API와 Google maps을 이용한 다양한 교통 정보 추출, 분석, 시각화

- 파이썬을 이용해 사용할 수 있는 Google maps 표현 방법 중 하나인 marker를 사용한다.
- 위의 path를 구하는 방법과 동일하게 marker의 색과 위치를 지정하고 urlstr에 추가한다. marker의 번호도 지정가능한데, 이는 xml 문서에 나온 순서대로 0~9, A~Z로 한다.

```
if p<10:
    markers+= '&markers=size:mid%7Ccolor:' + color[p]+ '%7Clabel:' + str(p) + '%7C' + mylist[p][0]
else:
    markers+= '&markers=size:mid%7Ccolor:' + color[p]+ '%7Clabel:' + chr(cnt) + '%7C' + mylist[p][0]
    cnt += 1
```

- path와 marker는 동시에 지정할 수 있으며, 적용한 결과 다음과 같다. 추가적으로 raodmap 형식을 hybrid 형식으로 변경해 사실성을 높였다.




결론 - tmap API를 통해 얻은 교통 정보를 xml 형식의 파일로 다운받고, 이 중 사용하려는 정보만을 추출해낼 수 있다. 또한 추출된 정보를 Google maps의 표현 방법으로 나타내는 것이 가능하며, 중간에 tmap 정보를 google 정보 형식으로 변환하는 과정이 있다.

시각화 방법에는 여러 가지가 있으나, 파이썬으로 표현하는 데에는 한계가 있으며, 따라서 javascript가 대중적으로 쓰인다.