

파이썬은 작업마다 필요한 모듈이 다르며, 상황에 따라 모듈끼리의 호환성 문제로 버전을 다운해야할  
상황들이 많음.

→ 작업별로 별도의 가상환경(envs)을 만들어서 작업하는것을 권장함.

파이썬만 설치한 경우보다 conda를 설치한경우가 가상환경 관리가 많이 쉬움

## 파이썬 가상환경작업 (가상환경의 개념알기)

<https://velog.io/@tngus0325/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EA%B0%80%EC%83%81%ED%99%98%EA%B2%BD-Conda%EC%99%80-Venv%EC%9D%98-%EC%9D%B4%ED%95%B4>

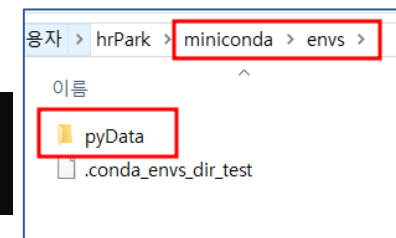
콘다를 설치한 가상환경세팅방법과 – conda create  
파이썬만 설치한 가상환경세팅방법이 다름 - venv

## 가상환경 세팅법, 이외 가상환경 제거등은 구글검색

(작업1) 아나콘다 환경에서 가상환경 생성, 파이썬 버전 설정 가능

가상환경이름

```
(base) C:\Users\WhrPark>conda create -n pyData python==3.8
```



(작업2) 가상환경으로 전환

```
(base) C:\Users\WhrPark>conda activate pyData  
(pyData) C:\Users\WhrPark>
```

(작업3) 생성된 가상환경 확인 :

```
(base) C:\Users\WhrPark>conda env list  
C:\Users\WhrPark\miniconda3\envs\test  
C:\Users\WhrPark\miniconda3\envs\test  
pyData  
C:\Users\WhrPark\miniconda3\envs\pyData  
C:\Users\WhrPark\miniconda3\envs\test
```

miniconda의  
envs폴더안에  
pyData가 생성되  
어야함.

(작업4) 가상환경에 모듈 설치하기

```
(pyData) C:\Users\WhrPark>conda install urllib3
```

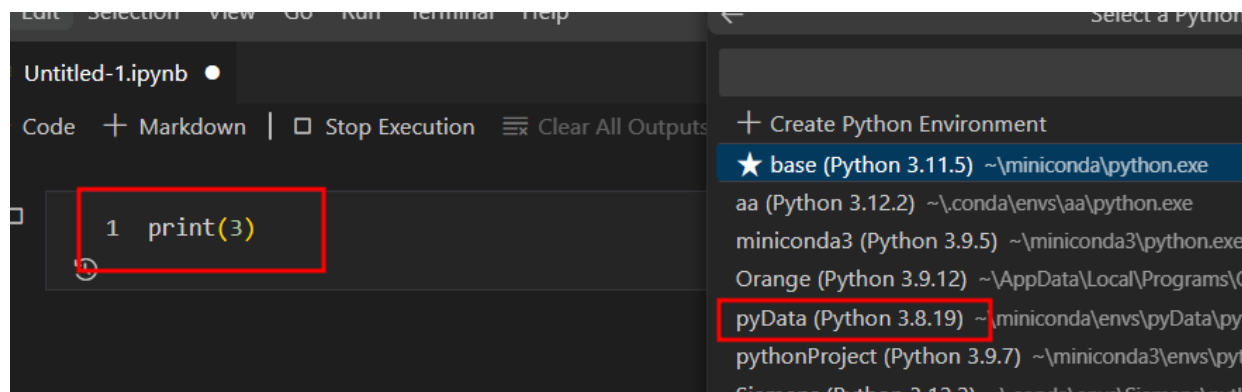
작업(교재 332)

교재의 sample url 다음으로 수정

url='http://openapi.seoul.go.kr:8088/sample/json/SeoulPublicLibraryInfo/1/5/'

vs코드에서 작업하고자함 , ipynb 하고 py작업 모두 가능하며 일단 ipynb로 작업  
ipynb(주피터노트북)은 가상환경을 새로 생성해서 처음작업시에 ipynb와 관련한 작업을 다시 설치하는 메세지 나  
옴. install 선택하여 새로 설치함

file-new-jupyter notebook  
실행하고  
코드 샘플후  
ctrl+enter해서  
파이썬 가상환경을  
pyData로 선택함.



#####

### 외부 url 읽기

### 서울시 도서관 정보 읽기(json으로 읽고자함)

### 샘플자료만 확인

#####

```
import urllib.request
```

```
import json
```

```
url='http://openapi.seoul.go.kr:8088/sample/json/SeoulPublicLibraryInfo/1/5/'
```

```
plainText=urllib.request.urlopen(url).read().decode()
```

```
jsonData=json.loads(plainText)
```

```
libData=jsonData.get('SeoulPublicLibraryInfo').get('row')
```

```
for ele in libData:
```

```
    name=ele.get('LBRRY_NAME')
```

```
    tel=ele.get('TEL_NO')
```

```
    addr=ele.get('ADRES')
```

```
    print(name,tel,addr)
```

GPT

다음작업 파이썬 함수로  
파이썬코드복사붙여넣기

GPT

함수를 더 쪼개서

#####

### 외부 url 읽기

### 서울시 도서관 정보 읽기(json으로 읽고자함)

### 샘플자료만 확인

#####

```
import urllib.request
```

```
import json
```

```
url='http://openapi.seoul.go.kr:8088/sample/json/SeoulPublicLibraryInfo/1/5/'
```

```
plainText=urllib.request.urlopen(url).read().decode()
```

```
jsonData=json.loads(plainText)
```

```
libData=jsonData.get('SeoulPublicLibraryInfo').get('row')
```

```
for ele in libData:
```

```
    name=ele.get('LBRRY_NAME')
```

```
    tel=ele.get('TEL_NO')
```

```
    addr=ele.get('ADRES')
```

```
    print(name,tel,addr)
```

GPT

다음작업 파이썬 함수로

파이썬코드복사붙여넣기

GPT

함수를 더 쪼개서

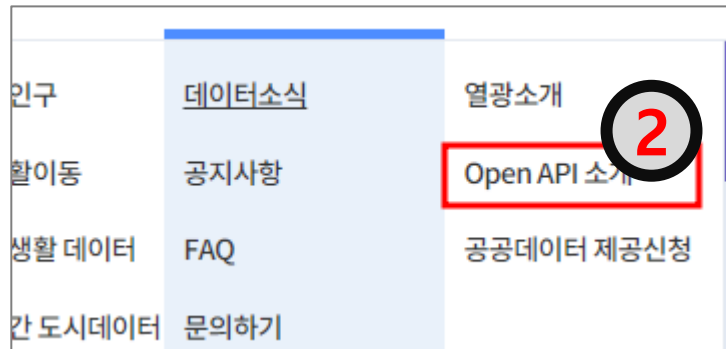
## [사전지식] 공공데이터란? open API란?

(1) 서울시 공공데이터에 접속 <https://data.seoul.go.kr/>

(2) 회원가입 및 openAPI키 발급받기

서울시 공공데이터

- 공공데이터
- OpenAPI



(3) 일반 인증키 신청

(4) 인증키관리에서 본인인증키 복사



- [1] 서울시 공공데이터
- [2] 검색어 입력 '공공도서관'
- [3] 공공데이터 클릭
- [4] 서울시 공공도서관 현황정보 클릭

The screenshot shows the '서울 열린데이터 광장' (Seoul Open Data Portal) homepage. The navigation bar includes '공공데이터' (Public Data) highlighted with a red box and a circled '1'. Below the header, a search bar contains '공공도서관' (Public Library) with a circled '2'. To the left, a sidebar lists categories like '문화/관광 (2)' and '교육 (1)'. The main content area shows search results for '공공도서관' with a circled '3' on the '공공데이터(3)' link. Below this, a link for '서울시 공공도서관 현황정보' (Seoul Public Library Status Information) is highlighted with a red box and a circled '4'. The page also displays a summary of 3 search results and a list of related tags.

- [5] OpenAPI
- [6] 샘플 URL 복사해서

The screenshot shows the '미리보기' (Preview) page for the 'Open API'. The 'Open API' tab is selected and highlighted with a red box and a circled '5'. Below the tabs, the '샘플 URL' (Sample URL) section is highlighted with a red box and a circled '6'. The URL is 'http://openapi.seoul.go.kr:8088/(인증키)/xml/SeoulPublicLibraryInfo/1/5/'. Below the URL, an '예제' (Example) section shows the XML response structure, including fields like 'list\_total\_count', 'RESULT', 'CODE', 'MESSAGE', 'LBRRY\_SEQ\_NO', and 'LBRRY\_NAME'.

## 제공되는 샘플

[http://openapi.seoul.go.kr:8088/\(인증키\)/xml/SeoulPublicLibraryInfo/1/5/](http://openapi.seoul.go.kr:8088/(인증키)/xml/SeoulPublicLibraryInfo/1/5/)

샘플인증키

78734b626c62616b35344947554c45

코드 변환 url을 상황에 따라 나누어서 변환해야 하는 경우가 많으므로 아래와 같이 url을 나누어 작업함

```
6 ✓ import urllib.request
7 import json
8
9 apiKey='78734b626c62616b35344947554c45'
10 url='http://openapi.seoul.go.kr:8088/'
11 url+=apiKey
12 url+=' /json/SeoulPublicLibraryInfo/1/5/'
13 print(url)
14
```

✓ 0.0s

전체데이터 갯수는 현재 206개이지만  
샘플로 5개만 확인함

<http://openapi.seoul.go.kr:8088/78734b626c62616b35344947554c45/json/SeoulPublicLibraryInfo/1/5/>

url이 잘 출력되는지 확인  
용



```
{
  "SeoulPublicLibraryInfo": {
    "list_total_count": 206,
    "RESULT": {
      "CODE": "INFO-000",
      "MESSAGE": "정상 처리되었습니다"
    },
    "row": [
      {
        "LBRRY_SEQ_NO": "1935",
        "LBRRY_NAME": "강남구립 못골도서관",
        "GU_CODE": "0008",
        "CODE_VALUE": "강남구",
        "ADRES": "서울특별시 강남구 자곡로 116"
      }
    ]
  }
}
```

```
import urllib.request
import json
```

```
api_Key='78734b626c62616b35344947554c45'
url='http://openapi.seoul.go.kr:8088/'
url+=api_Key
url+= '/json/SeoulPublicLibraryInfo/1/206/'
plainText=urllib.request.urlopen(url).read().decode()
jsonData=json.loads(plainText)
libData=jsonData.get('SeoulPublicLibraryInfo').get('row')
```

데이터셋의 갯수는 직접 확인하여서 206개로 변경해도 되고 계속 변동되는 자료라면 List\_total\_count값으로 다시 연결할수는 있지만 여기서는 갯수를 직접 입력하는 코드로 작업하고자 함.

```
import pandas as pd
df = pd.DataFrame(libData)
display(df)
```

```

1 import pandas as pd
2 df = pd.DataFrame(libData)
3 display(df)
4 df.csv()

```

0.0s

field라고 함.

	LBRRY_SEQ_NO	LBRRY_NAME	GU_CODE	CODE_VALUE	ADRES	TEL_NO	HMPG_URL	OP_TIME	FDRM_
0	1935	강남구립못골도서관	08	강남구	서울특별시 강남구 자곡로 116	02-459-5522	http://library.gangnam.go.kr/mglib	평일 : 09:00~22:00, 주말 : 09:00~22:00	매주 화

iloc[:,0]      iloc[:,1]      iloc[:,2]

df['TEL\_NO'] 하면 전화번호 필드 데이터만 출력됨  
df[['TEL\_NO','ADDRESS']] 하면 전화번호와 주소 필드가 출력됨

df.iloc[행번호,열번호], 행번호 열번호 위치에 : 이 있으면 모든 자료란 뜻

df.iloc[:,5] 는 전체행, 5열임 / 전화번호 필드 출력됨

df.iloc[:, [5,4]]는 전체행, 5열/4열 출력 (전화번호와 주소 필드 출력됨)

모든 자료가 아니라 필요한 항목만 변환하고자 한다면 (교재 332)

```
dataList=[]  
for item in libData:  
    name=item.get('LBRRY_NAME')  
    tel=item.get('TEL_NO')  
    addr=item.get('ADRES')  
    dataList.append({'name':name,  
                    'tel':tel,  
                    'addr':addr})
```

dataList

```
1 import pandas as pd  
2 pd.DataFrame(dataList)  
✓ 0.0s
```

2

	name	tel	addr
0	강남구립못골도서관	02-459-5522	서울특별시 강남구 자곡로 116
1	강남역삼푸른솔도서관	02-2051-1178	서울특별시 강남구 테헤란로8길 36. 4층
2	개포하늘꿈도서관	02-451-1511	서울특별시 강남구 개포로 110길 54
3	논현도서관	02-3443-7650	서울특별시 강남구 학동로43길 17(논현동) 논현2동주민센터 6층

```
9 dataList[:2]
```

✓ 0.0s

1

```
[{'name': '강남구립못골도서관', 'tel': '02-459-5522', 'addr': '서울특별시 강남구 자곡로 116'},  
{ 'name': '강남역삼푸른솔도서관',  
  'tel': '02-2051-1178',  
  'addr': '서울특별시 강남구 테헤란로8길 36. 4층'}]
```

## 참고

모든 자료가 아니라 필요한 항목만 변환하고자 한다면 (교재 332)

```
dataList=[]  
for item in libData:  
    name=item.get('LBRRY_NAME')  
    tel=item.get('TEL_NO')  
    addr=item.get('ADRES')  
    dataList.append({'name':name,  
                    'tel':tel,  
                    'addr':addr})
```

dataList

```
1 import pandas as pd  
2 pd.DataFrame(dataList)  
✓ 0.0s
```

2

	name	tel	addr
0	강남구립못골도서관	02-459-5522	서울특별시 강남구 자곡로 116
1	강남역삼푸른솔도서관	02-2051-1178	서울특별시 강남구 테헤란로8길 36. 4층
2	개포하늘꿈도서관	02-451-1511	서울특별시 강남구 개포로 110길 54
3	논현도서관	02-3443-7650	서울특별시 강남구 학동로43길 17(논현동) 논현2동주민센터 6층

```
9 dataList[:2]
```

✓ 0.0s

1

```
[{'name': '강남구립못골도서관', 'tel': '02-459-5522', 'addr': '서울특별시 강남구 자곡로 116'},  
{ 'name': '강남역삼푸른솔도서관',  
  'tel': '02-2051-1178',  
  'addr': '서울특별시 강남구 테헤란로8길 36. 4층'}]
```

1

```

1 libData=jsonData['SeoulPublicLibraryInfo']['row']
2                               위로 5개의 자료보기
3 import pandas as pd
4 df=pd.DataFrame(libData)
5 df.head() # = df.head(5)

```

-> df.head()는 5가 생략된 코  
드임  
df.head(3) 위로 3개의 자료보  
기

✓ 0.0s

LBRRY_SEQ_NO	LBRRY_NAME	GU_CODE	CODE_VALUE	ADRES
0	1925	강남구립못골	0008	강남구 강남대로 1925

서울특  
별시

3

```

1 #df[['TEL_NO','HMPG_URL']] # df.
2 df.iloc[:, [6,5]]

```

✓ 0.0s

HMPG

0	http://library.gangnam.go.k
---	-----------------------------

1	https://library.gangnam.go.kr/ysplib/i
---	--

2

```

1 df['TEL_NO'][:3] # df.iloc[:, 5][:3]

```

✓ 0.0s

0 02-459-5522

1 02-2051-1178

2 02-451-1511

Name: TEL\_NO, dtype: object

4

```

1 # 서울시 도서관 전체 갯수는?
2 len(df)

```

✓ 0.0s

206

5

```
1  🐞 서울시의 도서관이 있는 구의 갯수는?
2  len(df['CODE_VALUE'].unique()), df['CODE_VALUE'].unique()
```

```
✓ 0.0s
```

```
(25,
array(['강남구', '강동구', '강북구', '강서구', '관악구', '광진구', '구로구', '금천구', '노원구',
       '도봉구', '동대문구', '동작구', '마포구', '서대문구', '서초구', '성동구', '성북구',
       '양천구', '영등포구', '용산구', '은평구', '종로구', '중구', '중랑구'], dtype=object))
```

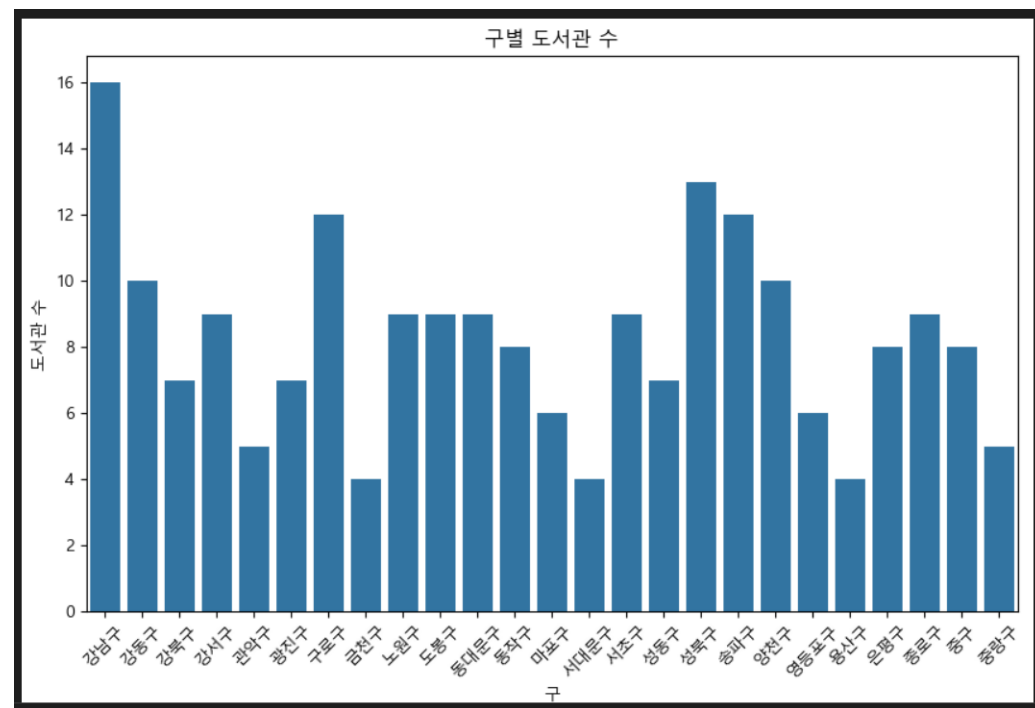
```
1  #!pip install seaborn
2  #!pip install matplotlib
3  import matplotlib.pyplot as plt
4  import seaborn as sns
5
6  # 파이썬 한글깨짐으로 검색
7  # https://giveme-happyending.tistory.com/168
8  plt.rcParams['font.family'] = 'Malgun Gothic'
9  plt.rcParams['axes.unicode_minus'] = False
10
11 plt.figure(figsize=(10, 6))
12 ax = sns.countplot(data=df, x='CODE_VALUE')
13
14 # x축 레이블 회전
15 ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
16 plt.title('구별 도서관 수')
17 plt.xlabel('구')
18 plt.ylabel('도서관 수')
19 plt.show()
```

강의때 작업하지 않음.

seaborn 차트는 matplotlib 차트를 기반으로 한 데이터분석용 차트임.

이 자료는 시분차트를 사용할만큼의 데이터가 없어서 효과적이지 않으나 seaborn 차트는 분석에 특화되어 있는 차트임.

sns.countplot(data=df, x='CODE\_VALUE')는 CODE\_VALUE를 unique(중복제거)해서 그 갯수를 세어서 차트를 그리는 기능임





6

```

1 # 각 구별로 도서관은 몇개가 있나요?
2 df['CODE_VALUE'].value_counts() # 숫자가 가장많은 값부터 출력
3
4 # 도서관 갯수가 많은 상위 top3 구는 어디인가요??
5 df['CODE_VALUE'].value_counts()[:3]

```

✓ 0.0s

```

CODE_VALUE
강남구      16
성북구      13
구로구      12

```

7

```

1 ### 강남구 데이터셋만 추출해주세요
2 df[df['CODE_VALUE']=='강남구']

```

✓ 0.0s

LBRRY_SEQ_NO	LBRRY_NAME	GU_CODE	CODE_VALUE
--------------	------------	---------	------------

0	1935	강남구립못골 도서관	0008	강남구
---	------	---------------	------	-----



## [사전알기] 딕셔너리 구조로 컬러맵 구성하기

```
1  ## 미리알기
2  ## 딕셔너리구조를 이용하여 컬러색상표를 제작할수 있음.
3  color_map = {
4      "강남구": "blue",
5      "강동구": "green",
6      "강서구": "red",
7      # 추가적인 라이브러리 이름과 색상을 추가하세요
8  }
9
10 color_map.get('강남구'),color_map.get('강서구')
11
```

✓ 0.0s

('blue', 'red')

## [사전알기] 지도 마커에 대하여

GPT에게

파이썬 folium의 아이콘 모양 변경 기본 샘플코드

GPT에게

아이콘 모양 종류와 적용법

GPT에게

마커가 아니라 동그라미로 표시하려면

```
#### 지도시각화
#### 구글에서 제공하는 지도맵을 세팅(출력) , 줌인아웃 사이트, 중심좌표
#### 내가 표시하고자 하는 위경도값을 그 구글맵지도에 추가한다.
```

```
#!pip install folium
import folium
```

```
# 위도, 경도
latitude = 37.48283
longitude = 126.8830
```

```
m = folium.Map(location=[latitude, longitude],
                zoom_start=10,
                width=750,
                height=500
                )
```

```
# 반드시 tmp 변수를 작성해야 하는것은 아님. df 자료로 작업해 됨.
tmp=df[['CODE_VALUE','XCNTS','YDNTS','LBRRY_NAME','OP_TIME']]
```

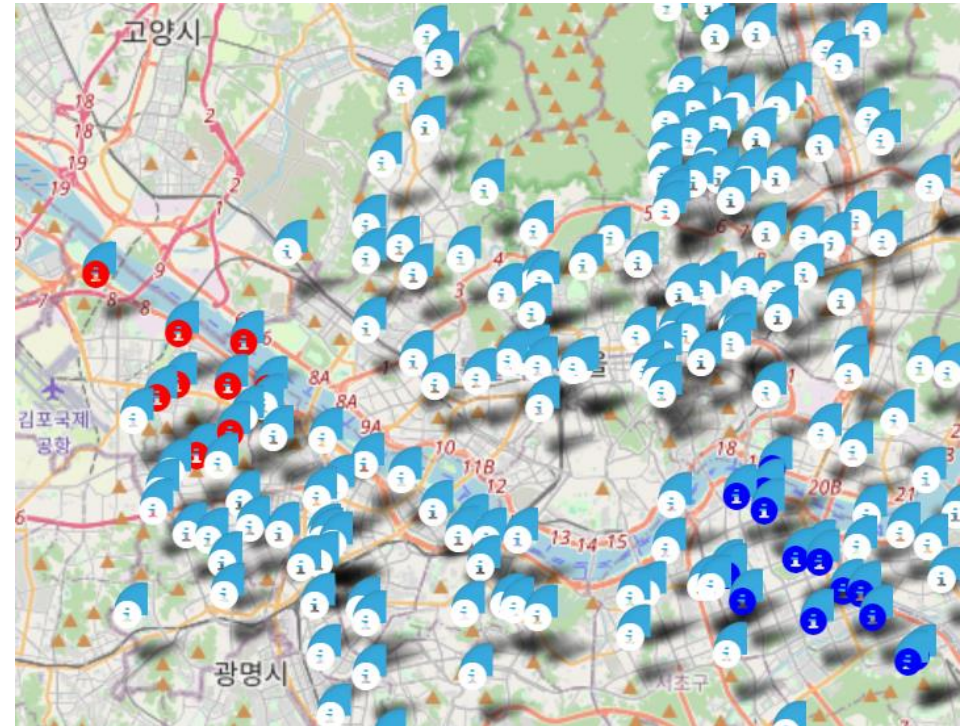
```
# LBRRY_NAME에 따른 색상 매핑을 위한 딕셔너리
color_map = {
    "강남구": "blue",
    "강동구": "green",
    "강서구": "red",
    # 추가적인 라이브러리 이름과 색상을 추가하세요
}
```

```
# 지역별 색상을 다르게 하기 위하여서 CODE_VALUE값 추가함.
```

```
for gu,lat, lon,name,time in zip(tmp['CODE_VALUE'],tmp['XCNTS'], tmp['YDNTS'],tmp['LBRRY_NAME'],tmp['OP_TIME']):
    pop=name+time
    color = color_map.get(gu)
```

```
    folium.Marker([lat, lon],
                  popup=pop,
                  icon=folium.Icon(icon_color=color, icon_size=(20,20)) # gpt 검색 , 파이썬 지도 차트 아이콘 크기
                  ).add_to(m)
```

```
m
```



```

#### 지도시각화
#### 구글에서 제공하는 지도맵을 세팅(출력) , 줌인아웃 사이트, 중심좌표
#### 내가 표시하고자 하는 위경도값을 그 구글맵지도에 추가한다.
#!pip install folium
import folium

```

```

# 위도, 경도
latitude = 37.48283
longitude = 126.8830

```

```

m = folium.Map(location=[latitude, longitude],
                zoom_start=10,
                width=750,
                height=500
            )

```

```

# 반드시 tmp 변수를 작성해야 하는것은 아님. df 자료로 작업해 됨.
tmp=df[['CODE_VALUE','XCNTS','YDNTS','LBRRY_NAME','OP_TIME']]

```

```

# LBRRY_NAME에 따른 색상 매핑을 위한 딕셔너리
color_map = {
    "강남구": "blue",
    "강동구": "green",
    "강서구": "red",
    # 추가적인 라이브러리 이름과 색상을 추가하세요
}

```

```

# 지역별 색상을 다르게 하기 위하여서 CODE_VALUE값 추가함.
for gu,lat, lon,name,time in zip(tmp['CODE_VALUE'],tmp['XCNTS'], tmp['YDNTS'],tmp['LBRRY_NAME'],tmp['OP_TIME']):
    pop=name+time
    color = color_map.get(gu)

```

```

folium.CircleMarker([lat, lon],
                    popup=pop,
                    radius=5, # 원의 반지름
                    color=color,
                    fill=True,
                    fill_color=color,
                    fill_opacity=1 #투명도

                    ).add_to(m)

```

```

m

```

