



[1차\_14주(6월 2일) 화상 강의 시간에 개별 간단 발표]

- 1. 분석하고 싶은 주제와 연구문제(5개이상) 작성
- 2. 분석할 데이터 확보와 파일 정리 (데이터는 위도와 경도가 있는 공공데이터 활용, 클롤링 등 자유롭게 구하기, 13주 화상강의 시간에 제공된 파일 이용도 가능)

[2차\_15주 기말고사 기간(6월 7일 ~ 6월 13일 24:00:00) 작업]

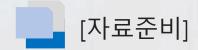
- 3. 데이터 분석 및 지도 활용 시각화
- 4. 자료 해석

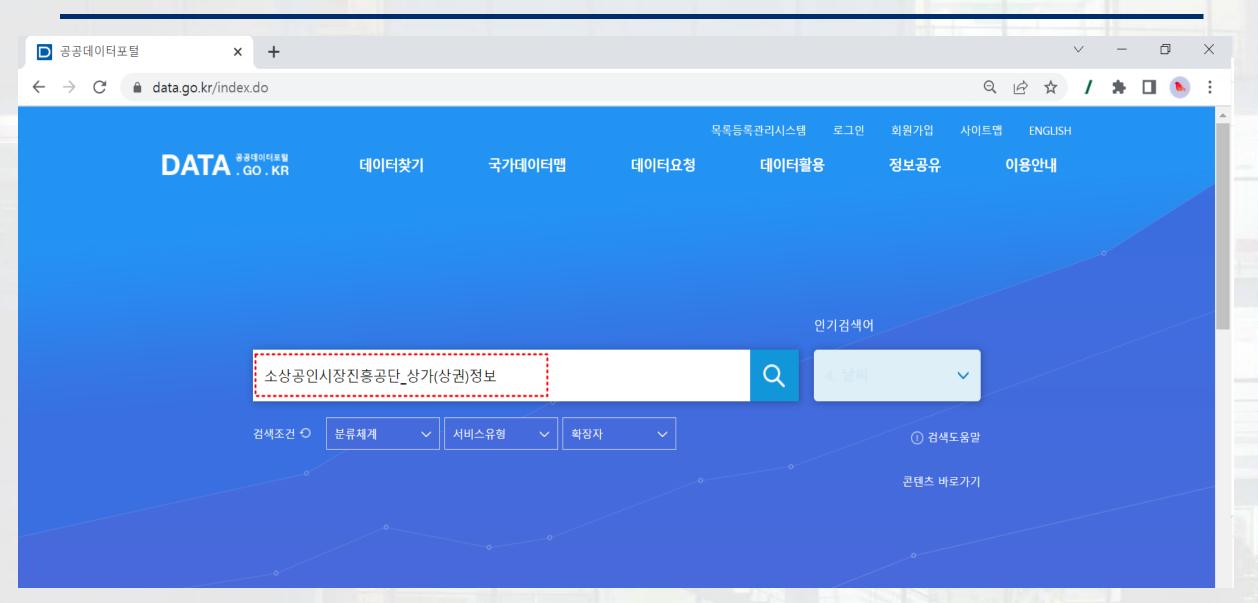


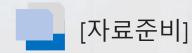
# 오늘의 학습

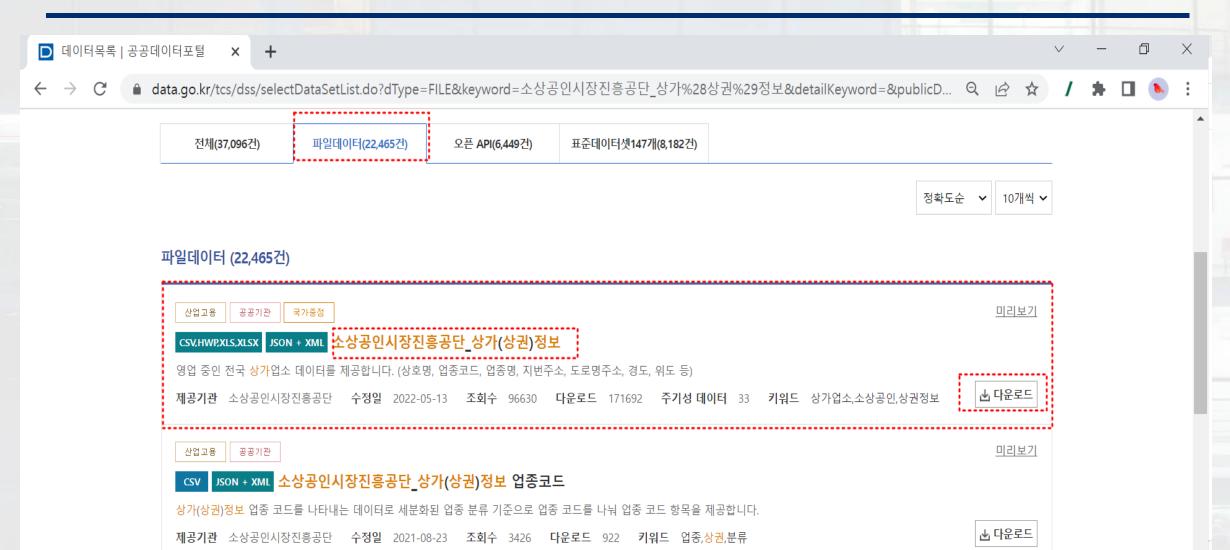
# 학습내용

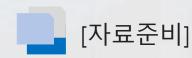
- 지도를 만들기 위한 자료 준비
- 지도 기반 데이터 분석 및 시각화(Python의 Folium 패키지)
- numpy 이해





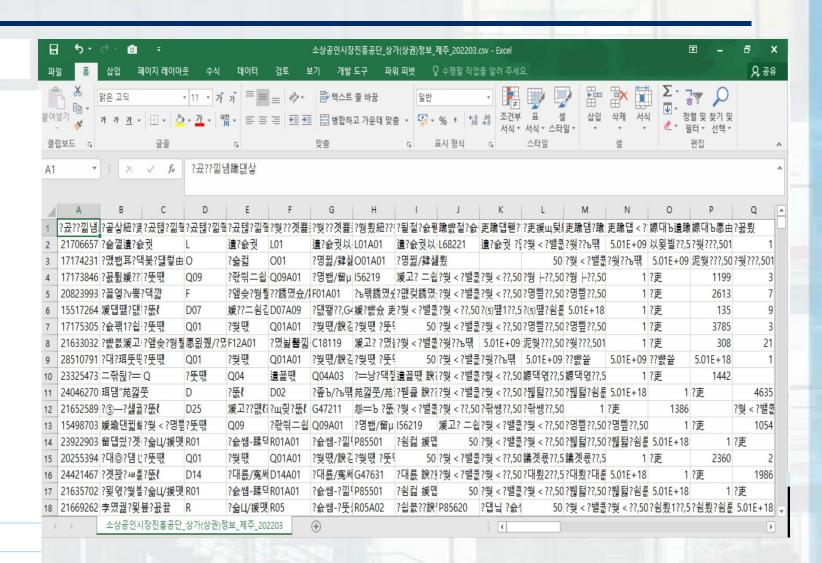






#### 📆 소상공인시장진흥공단\_상가(상권)정보.zip

- [필독]파일열람방법.txt
- 🛍 소상공인시장진흥공단\_상가(상권)정보\_강원\_202203.csv
- 🛍 소상공인시장진흥공단\_상가(상권)정보\_경기\_202203.csv
- 💶 소상공인시장진흥공단 상가(상권)정보 경남 202203.csv
- 💶 소상공인시장진흥공단 상가(상권)정보 경북 202203.csv
- 🛍 소상공인시장진흥공단\_상가(상권)정보\_광주\_202203.csv
- 💶 소상공인시장진흥공단 상가(상권)정보 대구 202203.csv
- 🛍 소상공인시장진흥공단 상가(상권)정보 대전 202203.csv
- 🛍 소상공인시장진흥공단\_상가(상권)정보\_부산\_202203.csv
- 💶 소상공인시장진흥공단 상가(상권)정보 서울 202203.csv
- 뢻 소상공인시장진흥공단\_상가(상권)정보\_세종\_202203.csv
- 💵 소상공인시장진흥공단\_상가(상권)정보\_울산\_202203.csv
- 녧 소상공인시장진흥공단\_상가(상권)정보\_인천\_202203.csv
- 💶 소상공인시장진흥공단 상가(상권)정보 전남 202203.csv
- 소상공인시장진흥공단 상가(상권)정보 전북 202203.csv
- 🔬 소상공인시장진흥공단\_상가(상권)정보\_제주\_202203.csv
- ▲ 소상공인시장진흥공단 상가(상권)정보 충남 202203.csv
- ▲ 소상공인시장진흥공단 상가(상권)정보 충북 202203.csv





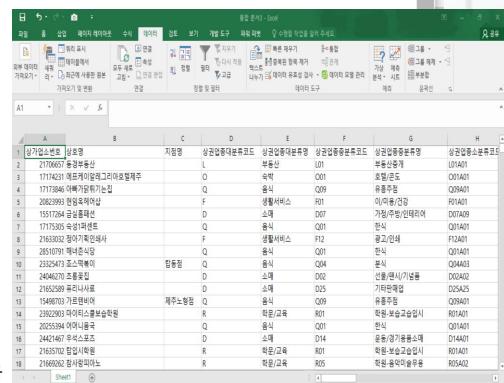
#### [자료준비]

🧻 [필독]파일열람방법.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

<MS엑셀(Excel)에서 UTF-8로 된 CSV파일 불러오기>

- 기존 euc-kr 데이터 추출형식으로 인해 업소명 오류가 발견되었으며 이를 개선하고자 UTF-8 형식으로 추출하였습니다.
- 개발자가 아닌 일반사용자의 경우 euc-kr(UTF-8 이나 UTF-16)이 아닌 CSV파일을 엑셀에서 바로 열면 한글이 모두 깨지게 됩니다.
- 바로 열지 마시고 다음 절차를 거치시기 바랍니다.
- 엑셀을 실행하고 데이터 → 텍스트를 선택합니다.
- 가져올 파일을 선택하고 확인을 클릭합니다.
- 콤보 박스에서 적절한 코드 타입이 선택됐는지 확인합니다.
- \* UTF-8의 코드 페이지 넘버는 65001입니다.
- 구분 기호는 '쉼표' 선택합니다.
- \* 기본이 쉼표로 선택되어 있음
- ※ 업종(대분류): 관광/여가/오락, 부동산, 생활서비스, 소매, 숙박, 스포츠, 음식, 학문/교육



?좏씎二십 Q09A01 ?명봽/留µ I56219 漢고? 二십?쒖 < ?밷쿲?뭦 <

?슔쎕-珠<sup>다</sup>R01A01 ?슔쎕-?낆<sup>4</sup>P85501 ?쇰컲 援맵

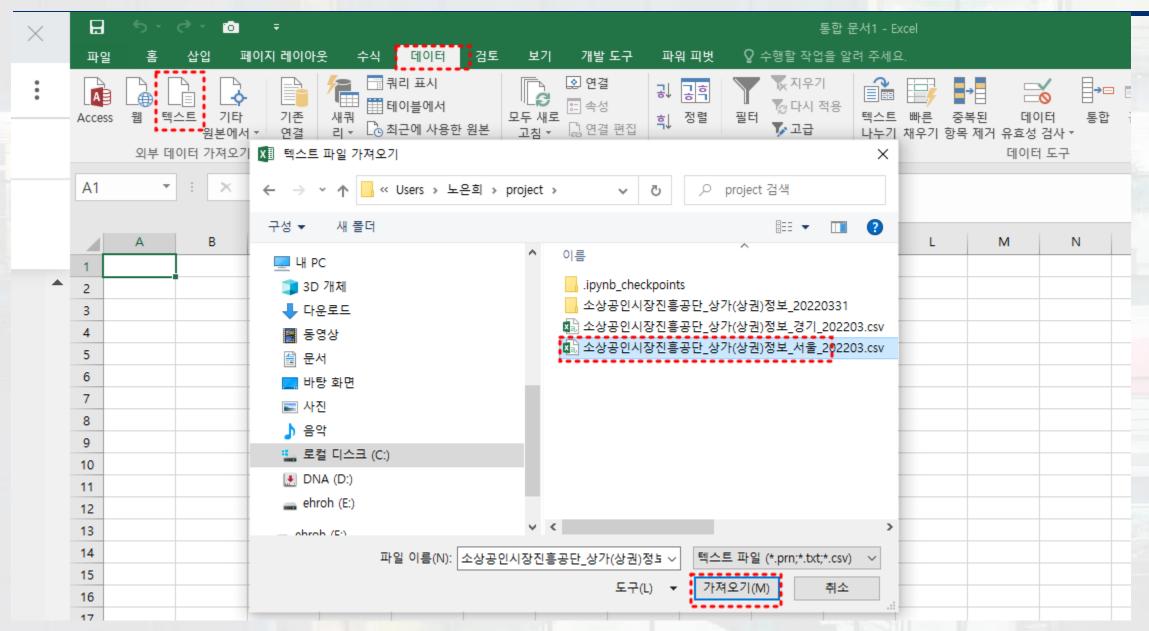
?대료/寃세D14A01 ?대료/寃세G47631 ?대료 飲???쒗 < ?별쿱?쒗 < ??,50?대됬2??,5?대됬2?1륜 5.01E+18

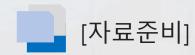
?좊Ъ/?ь떆苑갫쭛/苑 ?붿큪 按???쒖 < ?벨쿲?쒖 < ??,50 ?붼턿??,50 ?붼턿?4륜 5.01E+18

50 ?뛎 < ?밸큔?뛎 < ??,50 ?뛚턺??,50 ?뭲턺?쇰륜 5.01E+18

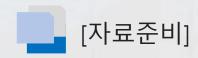


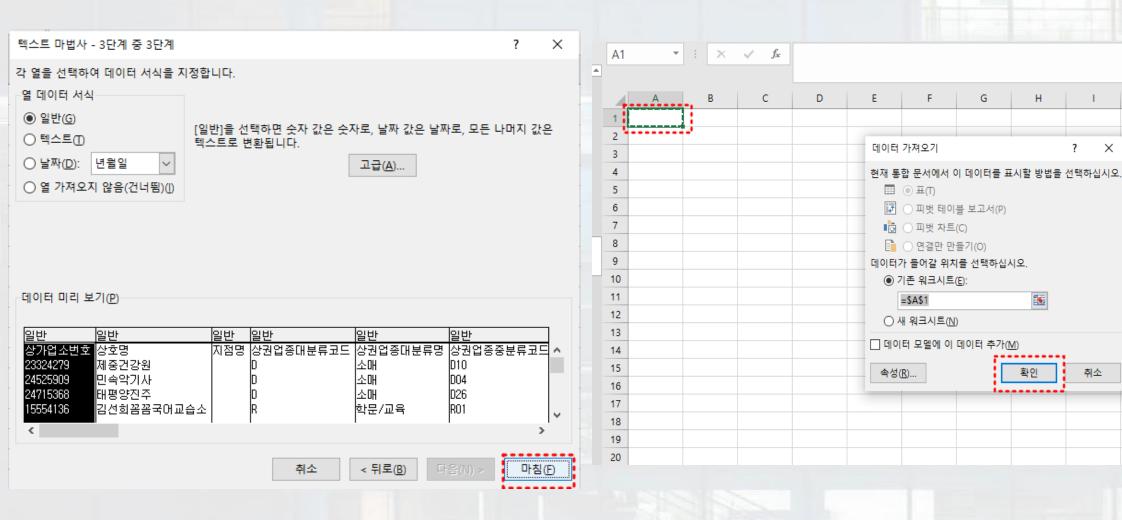
#### [자료준비]





| 텍스트 마법사 - 3단계 중 1단계 <b>?</b> ×   | 텍스트 마법사 - 3단계 중 2단계 <b>?</b> ×                                 |
|--|--|
| 데이터가 구분 기호로 분리됨(으)로 설정되어 있습니다.<br>데이터 형식이 올바로 선택되었다면 [다음] 단추를 누르고, 아닐 경우 적절하게 선택하십시오.<br>원본 데이터 형식   | 데이터의 구분 기호를 설정합니다. 미리 보기 상자에서 적용된 텍스트를 볼 수 있습니다.               |
| 원본 데이터의 파일 유형을 선택하십시오. <ul> <li>● 구분 기호로 분리됨(D)</li> <li>- 각 필드가 쉼표나 탭과 같은 문자로 나누어져 있습니다.</li> <li>○ 너비가 일정함(W)</li> <li>- 각 필드가 일정한 너비로 정렬되어 있습니다.</li> </ul>   | □ 탭(T) □ 세미콜록(M) □ 연속된 구분 기호를 하나로 처리(R) □ 심표(C) □ 공백(S)        |
| 구분 시작 행(R): 1 ♣ 원본 파일(O): 65001 : 유니코드(UTF-8) ✓  | □ 기타( <u>O</u> ):  |
| □ 내 데이터에 머리글 표시( <u>M</u> )  |  |
| C:₩Users₩노은희₩project₩소상공인시장진흥공단_상가(상권)정보_서울_202203.csv 파일 미리 보기  | 데이터 미리 보기(만)   |
| 1 상가업소번호,상호명,지점명,상권업종대분류코드,상권업종대분류명,상권업종중분류코드,상권업종중봉수<br>2 "23324279","제중건강원",,"D","소매","D10","건강/미용식품","D10A07","건강원","G47216","건강보호<br>3 "24525909","민속악기사",,"D","소매","D04","취미/오락관련소매","D04A09","악기판매","G47593","역<br>4 "24715368","태평양진주",,"D","소매","D26","시계/귀금속소매","D26A01","시계/귀금속","G47830","<br>5 "15554136","김선희꼼꼼국어교습소",,"R","학문/교육","R01","학원-보습교습입시","R01A01","학원-입 | 상가업소번호 상호명 지점명 상권업종대분류코드 상권업종대분류명 상권업종중분류코드 ^ 23324279 제중건강원 D |
| 취소 (뒤로(B) 다음(N) > 마침(F)  | 수 지로(B) 다음(N) > 마침(F)  |





# 지리 정보 분석 실습\_제주도 상권분석 및 시각화



- 설치 : pip install folium
- Anaconda Powershell Prompt (Anaconda3)

(base) PS C:₩Users₩노은희> pip install folium

- 설치 확인 : pip list
- Anaconda Powershell Prompt (Anaconda3)

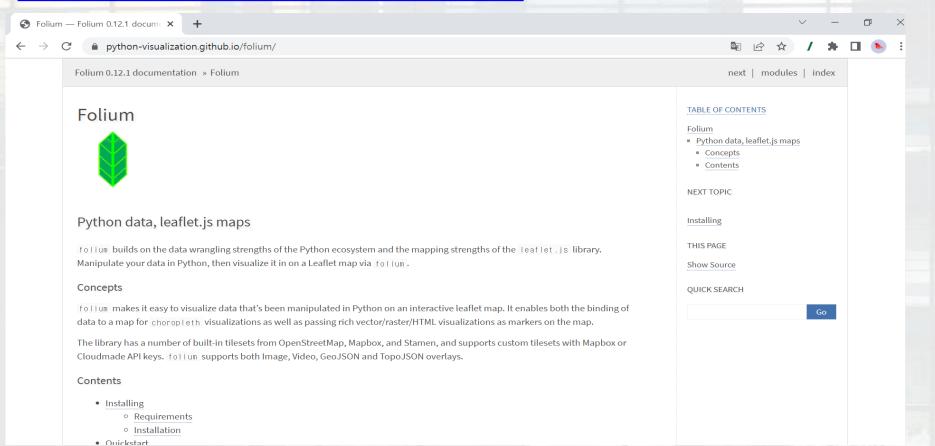
| (base) PS C:#Users#노은희>                                 | ⊳ pip list            |
|---|-----------------------|
| Package   | Version               |
| alabaster   | 0.7.12                |
| anaconda-client   | 1.9.0                 |
| anaconda-navigator                                      | 2.1.2                 |
| anaconda-project  | 0.10.1                |
| anyio   | 2.2.0                 |
| Anaconda Powershell Prompt (Anaconda Powershell Prompt) | conda3)               |
| decorator<br>defusedxml                                 | 5.1.0<br>0.7.1        |
| diff-match-patch<br>distributed                         | 20200713<br>2021.10.0 |

docutils 0.17.1 5.3.0 enable entrypoints et-xmlfile fastcache ilelock flake8 lask 0.12.1.post1 folium fonttools fsspec 2021.10.1 future 21.8.0 gevent



- location이라는 속성을 이용해 위도와 경도를 기본으로 입력
- 해당 위치를 중심으로 한 지도가 생성

아래 Github에서 더 많은 정보를 얻을 수 있다. <a href="https://python-visualization.github.io/folium/">https://python-visualization.github.io/folium/</a>





#### 1. 기본지도 생성하기

- location이라는 속성을 이용해 위도와 경도를 기본으로 입력
- 해당 위치를 중심으로 한 지도가 생성
  - 에 중 귀시를 중심으로 한 시로가 중중 # 기본 지도 생성 import folium map = folium.Map(location=[33.5137542483577,126.5368466]) #/ocation에 위도, 경도 정보를 알려, 제주특별자치도 제주시 동문로 96 # 기본 지도 생성 import folium map = folium.Map(location=[33.5137542483577,126.5368466]) map





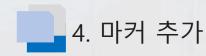
### 2. zoom\_start 속성\_지도 크기 변경

```
# zoom_start 속성 사용
  import folium
  map = folium.Map(location=[33.5137542483577,126.5368466], zoom_start = 15)# zoom_start 정보를 지정하여 확대, 최대 18
+
                                                                             사라봉길
                                                                                     국립제주
                                                                         1132
```



#### 3. 지도의 테마 변경

```
# tiles 옵션 변경을 통해 지도의 테마 변경, 지도의 기본 테마(tiles)는 OpenStreetMap으로 설정
2 import folium
map = folium.Map(location=[33.5137542483577,126.5368466], zoom_start = 15, tiles = 'Stamen Toner')
```



- •location: 마커를 추가할 위도/경도 좌표를 입력 후
- •popup: 표기할 팝업 문구 지정 (마우스 클릭시 표기되는 문구)
- •tooltip: 표기할 툴팁 지정 (마우스 오버시 표기되는 문구)

마커를 생성 후 기존에 생성된 지도 map에 추가

# 마커 표시하기 import folium = folium.Map(location=[33.51375 import folium folium.Marker(!location=[33.513754248

map = folium.Map(location=[33.5137542483577,126.5368466], zoom start = 15)

folium.Marker(location=[33.5137542483577,126.5368466], popup="세흥장식",

tooltip="제주특별자치도 제주시 동문로").add\_to(map)

map



# 5. 아이콘 변경

# 아이콘 변경 import folium

map = folium.Map(location=[33.5137542483577,126.5368466], zoom\_start = 15)

folium.Marker(location=[33.5137542483577,126.5368466], popup="세흥장식", tooltip="제주특별자치도 제주시 동문로",

icon = folium.lcon('red', icon='star')).add\_to(map) #folium.lcon(color=?, icon=?) 아이콘 변경

map





#### 6. [개인 실습] 학교 위도,경도 찾아서 지도에 표시하기

```
# popupOl나 tooltip에 html 코드를 삽입하여 이미지를 표기하거나 YouTube 영상도 삽입
   import folium
   map = folium.Map(location=[37.581471,127.010103], zoom_start = 16)
   folium.Marker([37_581471.127_0101031]
popup='<iframe width="560" height="315" src="https://www.hansung.ac.kr"></iframe>'
tooltip="한성대",
                icon=folium.lcon('red', icon='star')).add_to(map)
10 map
            혜화동
                                                                                       LANG
```

# 7. 제주 상

#### 7. 제주 상권 정보 파일 가져오기

```
# 7. 제주 삼권 정보 파일 가져오기
   import pandas as pd
   import warnings
                                       # 파이썬 코딩시 warnings이뜨면 무시
   warnings.filterwarnings('ignore') # 파이썬 코딩시 warnings이뜨면 무시
   df = pd.read_csv('소상공인시장진흥공단_상가(상권)정보_제주_202203.csv')
 8 | df.head(5)
                    상권업종대 분류코드
                            상 권 업 종 중 분 류 코 드
                                상 권 업 종 중 분 류 명
                        상권 업종 대 분류명
                                                                                          도로명주
                                     상권업
종소분
류코드
                                                 표준산
                                                                                                                                  경도
                                                 업분류
                                                                                                        번호
                                                                                                               보
                                                                                                                    보
                                                                                                번호
                                                  코드
                                                                                          치
           동
                                .
동
산
                                                                                          서
귀
포
           부
                                                                                    레코
                        동
                           L01
                                    L01A01
                                                L68221 ...
0 21706657
               NaN
                                                          5013010100102900005031397
                                                                                              697843 63591.0 NaN NaN NaN 126.561556 33
                                 중
                                개
                                                                                         70-
```



#### 8. 제주 상권 정보 파일에서 컬럼명 살펴보기

#### # 8. 제주 상권 정보 파일에서 컬럼명 살펴보기 2 pd.Series(df.columns) 상가업소번호 상호명 지점명 0 2 상권업종대분류코드 상권업종대분류명 5 6 상권업종중분류코드 상권업종소분류코드 상권업종소분류명 8 표준산업분류코드 9 표준산업분류명 10 11 시도코드 시도명 12 시군구코드 시군구명 행정동코드 13 14 15 행정동명 16 17 법정동코드 법정동명 지번코드 18 19 20 대지구분코드 대지구분명 21 22 23 24 25 26 치번본먼지 지번부번지 지번주소 도로명코드 도로명 건물본번지 27 28 29 건물부번지 건물관리번호 30 31 32 33 34 35 36 37 건물명 도로 중 구우편번호 구 신 무 편 동 정 보 총 청 호 호 경도 위도

dtype: object

제주 상권 정보 파일에서 컬럼명 살펴보기 pd.Series(df.columns)

#### 9. 필요한 컬럼 정보만 가져오기

# 0. 필요한 컬럼 정보만 가져오기

jeju = pd.read\_csv('소상공인시장진흥공단\_상가(상권)정보\_제주\_202203.csv')

5 # 필요한 컬럼 정보만 가져옵니다

i jeju\_sub = jeju[['시군구명', '상권업종대분류명', '상권업종중분류명', '위도', '경도','도로명','행정동명']] ' jeju\_sub.head(10)

|   | 시군구명 | 상권업종대분류명 | 상권업종중분류명   | 위도        | 경도         | 도로명                   | 행정동명 |
|---|------|----------|------------|-----------|------------|-----------------------|------|
| 0 | 서귀포시 | 부동산      | 부동산증개      | 33.250877 | 126.561556 | 제주특별자치도 서귀포시 중앙로      | 중앙동  |
| 1 | 서귀포시 | 숙박       | 호텔/콘도      | 33.248179 | 126.559015 | 제주특별자치도 서귀포시 중앙로47번길  | 천지동  |
| 2 | 제주시  | 음식       | 유흥주점       | 33.412363 | 126.264860 | 제주특별자치도 제주시 한림읍 한림중앙로 | 한림읍  |
| 3 | 제주시  | 생활서비스    | 이/미용/건강    | 33.485978 | 126.474887 | 제주특별자치도 제주시 정존11길     | 노형동  |
| 4 | 제주시  | 소매       | 가정/주방/인테리어 | 33.511339 | 126.517956 | 제주특별자치도 제주시 서문로4길     | 용담1동 |
| 5 | 제주시  | 음식       | 한식         | 33.483335 | 126.479894 | 제주특별자치도 제주시 노형10길     | 노형동  |
| 6 | 서귀포시 | 생활서비스    | 광고/인쇄      | 33.250127 | 126.559047 | 제주특별자치도 서귀포시 서문로29번길  | 천지동  |
| 7 | 서귀포시 | 음식       | 한식         | 33.166521 | 126.274126 | 제주특별자치도 서귀포시 대정읍 가파로  | 대정읍  |
| 8 | 제주시  | 음식       | 분식         | 33.516737 | 126.526797 | 제주특별자치도 제주시 임항로       | 건입동  |
| 9 | 제주시  | 소매       | 선물/팬시/기념품  | 33.516388 | 126.559265 | 제주특별자치도 제주시 일주동로      | 화북동  |



## 10. isin() DataFrame이나 Series에 존재 하는지를 True(존재), False(존재안함)로 반환

|    | 1 # 10. isin() DataFrameOl나 Series에 존재 하는지를 True(존재), False(존재안함)로 반환<br>2 sub_df = df.loc[df['행정동명'].isin(['건입동','일도이동'])] # isin() Pandas에서는 어떤 list에 존재하는 요소가 대상 DataFr<br>3 sub_df.head(5)<br>4 |            |             |         |                   |                 |                  |                 |                   |              |                  |                          |         |                            |           |           |         |         |         |              |
|----|---|------------|-------------|---------|-------------------|-----------------|------------------|-----------------|-------------------|--------------|------------------|--------------------------|---------|----------------------------|-----------|-----------|---------|---------|---------|--------------|
|    |   | 상가업소<br>번호 | 상<br>호<br>명 | 지점<br>명 | 상 권 업 종 대 분 류 코 드 | 상 권 업 종 대 분 류 명 | 상권 업 종 중 분 류 코 드 | 상 권 업 종 중 분 류 명 | 상권업<br>종소분<br>류코드 | 상권 업 종소 분류 명 | 표준산<br>업분류<br>코드 | 건물관리번호                   | 건물<br>명 | 도<br>로<br>명<br>주<br>소      | 구우편<br>번호 | 신우편<br>번호 | 동정<br>보 | 충정<br>보 | 호정<br>보 | 경도           |
|    | 8   | 23325473   | 죠 스떡 볶이     | 탑동점     | Q                 | 음식              | Q04              | 분식              | Q04A03            | 떡 볶<br>이 전 문 | 156194           | 501101070011442000202050 | 0 NaN   | 제주특별자치도제주시임항로<br>17        | 690801    | 63276.0   | NaN     | NaN     | NaN     | 126.526797 3 |
| 15 | 58  | 20492130   | 노마          | NaN     | Q                 | 음<br>식          | Q01              | 한<br>식          | Q01A01            | 한사 백 번 한정식   | I56111           | 501101070011090000103250 | 6 NaN   | 제 주 특별 자 치 도 제 주 시 연 무 정 드 | 690801    | 63282.0   | NaN     | 1.0     | NaN     | 126.535475 3 |



### 11. 행정동명이 건입동, 일도이동 행정명 위도, 경도

1 # 11. 행정들명이 건입동, 일도이동 행정명 위도, 경도 2 sub\_df[['위도', '경도', '상호명','행정동명']] #행정들명이 건입동, 일도이동 행정명 위도, 경도

|       | 위도        | 경도         | 상호명    | 행정동명 |
|-------|-----------|------------|--------|------|
| 8     | 33.516737 | 126.526797 | 죠스떡볶이  | 건입동  |
| 158   | 33.514009 | 126.535475 | 돈물     | 건입동  |
| 201   | 33.516208 | 126.530885 | 하나로국밥  | 건입동  |
| 224   | 33.514022 | 126.536648 | 백합문구사  | 건입동  |
| 244   | 33.517552 | 126.539323 | 한라종합가구 | 건입동  |
| •••   |           |            | •••    |      |
| 48146 | 33.516160 | 126.526107 | 드레곤    | 건입동  |
| 48378 | 33.516713 | 126.537712 | 승도조명   | 건입동  |
| 48485 | 33.516983 | 126.530054 | 정수산    | 건입동  |
| 48640 | 33.518007 | 126.535817 | 노픈커피   | 건입동  |
| 48661 | 33.507292 | 126.553874 | 건입부동산  | 건입동  |

540 rows × 4 columns

#### 12. MarkerCluster



#### 13. 가까운 거리의 Marker들을 군집시켜서 해당 건수를 표현

```
# 13. 가까운 거리의 Marker들을 군집시켜서 해당 권수를 표현

from folium.plugins import MarkerCluster

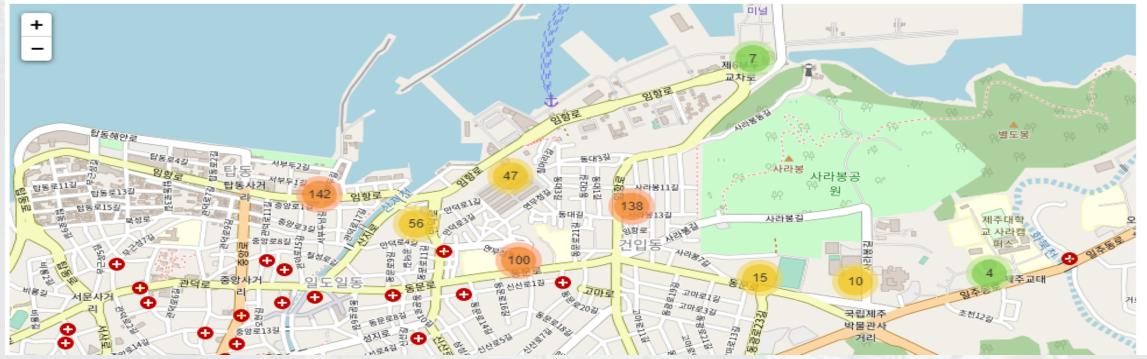
m = folium.Map(location=[33.5137542483577,126.5368466],zoom_start=15)

coords = sub_df[['위도', '경도']]

marker_cluster = MarkerCluster().add_to(m)

for lat, long in zip(coords['위도'], coords['경도']):
    folium.Marker([lat, long], icon = folium.lcon(color="green")).add_to(marker_cluster)

m
```



#### 14. 행정동영이 건입동, 일도이동 지역의 상권에서 커피점/카페 검색

- # 14. #행정들영이 건입동, 일도이동 지역의 상권에서 커피점/카페 검색
- | 3 | jeju\_coffee = sub\_df.loc[sub\_df['상권업종중분류명'] == '커피점/카페'] | 4 | jeju\_coffee = jeju\_coffee[['위도', '경도', '상호명','행정동명','상권업종중분류명']] | #행정통영이 건입동, 일도이동 행정명 위도, 경도
- 5 jeju\_coffee

|       | 위도        | 경도         | 상호명          | 행정동명 | 상권업종중분류명 |
|-------|-----------|------------|--------------|------|----------|
| 13829 | 33.516215 | 126.530724 | 스타벅스         | 건입동  | 커피점/카페   |
| 14006 | 33.516201 | 126.525773 | 에이바우트커피      | 건입동  | 커피점/카페   |
| 17860 | 33.514545 | 126.532819 | 몽티죠          | 건입동  | 커피점/카페   |
| 19839 | 33.513910 | 126.535552 | 희            | 건입동  | 커피점/카페   |
| 20286 | 33.516009 | 126.527746 | 다올방          | 건입동  | 커피점/카페   |
| 20467 | 33.515663 | 126.539656 | 꾸까           | 건입동  | 커피점/카페   |
| 22027 | 33.514733 | 126.539532 | 오            | 건입동  | 커피점/카페   |
| 27446 | 33.514947 | 126.529421 | 동문뮤지엄카페      | 건입동  | 커피점/카페   |
| 27746 | 33.516441 | 126.535338 | 마이티커피        | 건입동  | 커피점/카페   |
| 31751 | 33.517368 | 126.527561 | 투썸플레이스휘슬락호텔점 | 건입동  | 커피점/카페   |
| 33056 | 33.514054 | 126.536314 | 유정           | 건입동  | 커피점/카페   |
| 33124 | 33.513786 | 126.547964 | 카페타리         | 건입동  | 커피점/카페   |
| 37285 | 33.513786 | 126.547964 | 씨제이프레시웨이카페겨를 | 건입동  | 커피점/카페   |
| 39349 | 33.515073 | 126.540850 | 카페렌          | 건입동  | 커피점/카페   |
| 41748 | 33.515146 | 126.529547 | 이퀼           | 건입동  | 커피점/카페   |
| 41881 | 33.516376 | 126.526039 | 펫아이준         | 건입동  | 커피점/카페   |
| 43097 | 33.517368 | 126.527561 | 장산다방         | 건입동  | 커피점/카페   |
| 43819 | 33.506628 | 126.547244 | 커피로스터스       | 건입동  | 커피점/카페   |
| 46502 | 33.516133 | 126.535999 | 타로피플         | 건입동  | 커피점/카페   |
| 48054 | 33 516633 | 126 535582 | 하미슾커피곤반      | 거인동  | 커피전/카페   |

#### 15. 행정동명이 건입동, 일도이동 지역의 상권에서 커피점/카페 검색

```
# 15. 행정동명이 건입동, 일도이동 지역의 삼권에서 커피점/카페 지도에 표시

from folium.plugins import MarkerCluster

m = folium.Map(location=[33.5137542483577,126.5368466],zoom_start=15)

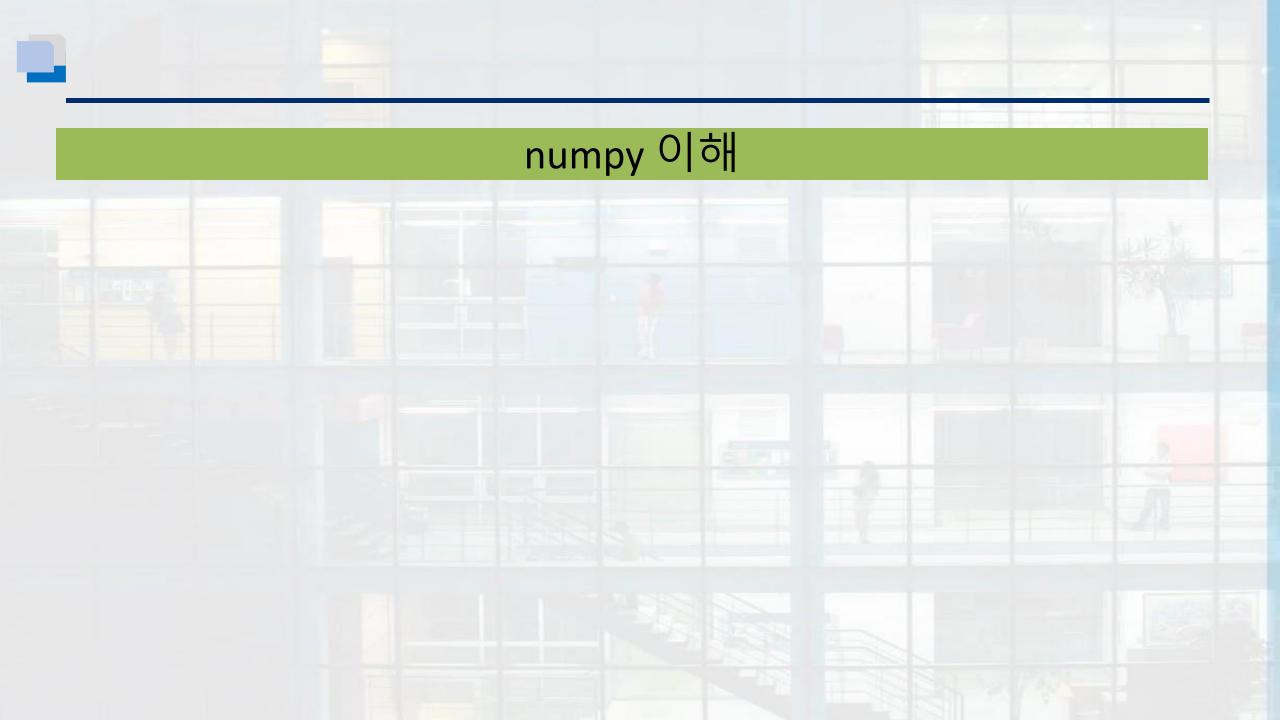
coords = jeju_coffee[['위도', '경도']]

marker_cluster = MarkerCluster().add_to(m)

for lat, long in zip(coords['위도'], coords['경도']): #zip()는 반복문 수행시 두개의 리스트 값을 묶어준다.
    folium.Marker([lat, long], icon = folium.lcon(color="green")).add_to(marker_cluster)

m
```





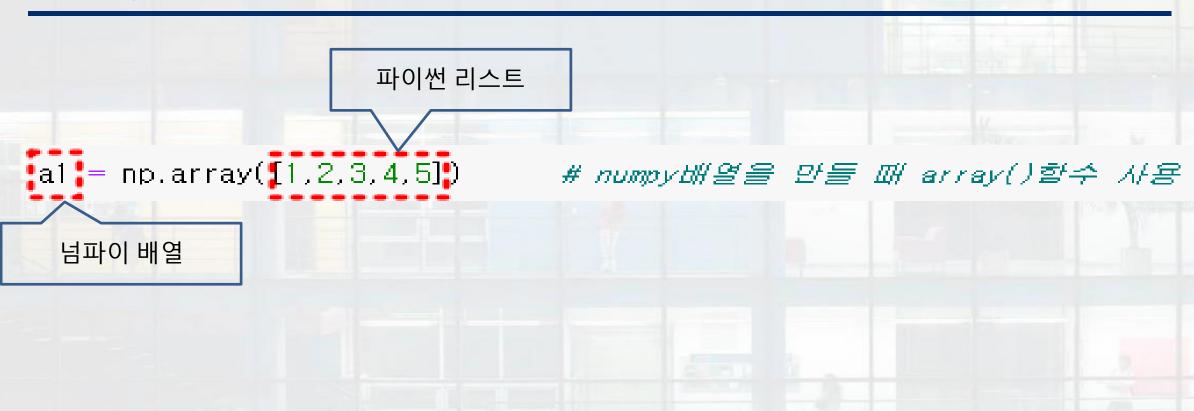


# ☑ 파이썬 list와 numpy 배열

- 배열은 동일한 자료형을 가진 데이터를 연속적으로 저장
- numpy는 ndarray객체 제공
- ndarray는 n차원 배열 의미하고 동일한 자료형의 항목들만 저장
- 파이썬의 list는 동일하지 않는 자료형도 저장 가능
- ndarray는 배열과 배열간의 수학적인 연산 가능
- ndarray는 풍부한 함수 제공



# numpy 배열 만들기



# numpy 배열 만들기

data type=> <U1

#### numpy 배열 생성하기 => 1차원 배열

```
### 시퀀스 데이터로부터 배열 생성
   import numpy as np # numpy 패키지 둘러오기
 3
   al = np.array([1,2,3,4,5]) # numpy배열을 만들 때 array()함수 사용
   print(al.dtype)
   print(a1)
int32
[1 2 3 4 5]
   data1 = [1.0, 2.0, 3.0, 4.0, 5.0]
   a2 = np.array(data1)
 3
   print(a2.dtype)
   print(a2)
float64
[1, 2, 3, 4, 5,]
  data2 = np.array(['1','2','3','4','5'])
  print(data2)
```



# numpy 배열 만들기

# numpy 배열 생성하기 ==> arange()

```
| ## arr = np.arange([start], stop[,step])|
  2 # start부터 시작해서 stop 전까지 step만큼 더해서 numpy 배열을 생성
  3 \mid b1 = np.arange(0,10,1)
  4 | b1
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
    | b2 = np.arange(0,10,2)|
 2 b2
array([0, 2, 4, 6, 8])
    | b3 = np.arange(5)|
  2 b3
array([0, 1, 2, 3, 4])
```



# numpy 배열 만들기\_다차원 배열

## numpy 배열 생성하기 ==> 다차원 배열

```
import numpy as np
c = np.array([[1,2],[3,4]]) # array()함수는 리스트를 인수로 받아서 배열을 생성, 2차원 배열 생성
c
array([[1, 2],
[3, 4]])
```



#### 배열 생성하기

## 배열생성하기 ==> reshape(m,n)을 추가하면 m x n 형태의 2차원 배열로 변경

```
import numpy as np
 | 2 | arr1 = np.arange(20).reshape(4,5) # reshape(m.n)을 추가하면 m x n 형태의 2차원 배열로 변경
 3 arr1
array([[ 0, 1, 2, 3, 4],
      [5, 6, 7, 8, 9],
      [10, 11, 12, 13, 14],
      [15, 16, 17, 18, 19]])
    arr2 = np.arange(30).reshape(3.10)
 2 arr2
array([[ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
      [10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
      [20, 21, 22, 23, 24, 25, 26, 27, 28, 29]])
    arr3 = np.zeros((5,5)) # zeros()함수는 0으로 재워진 넘파이 배열 생성
 2 arr3
array([[0., 0., 0., 0., 0.],
      [0., 0., 0., 0., 0.],
      [0., 0., 0., 0., 0.]
      [0...0..0..0..0..0.]
      [0...0..0..0..0..0.]
```

### 배열 생성하기

```
arr4 = np.ones((5,5)) # ones()함수는 1으로 제워진 남파이 배열 생성
    arr4
array([[1., 1., 1., 1., 1.],
      [1., 1., 1., 1., 1.],
      [1., 1., 1., 1., 1.],
      [1., 1., 1., 1., 1.],
      [1., 1., 1., 1., 1.]])
    # 냄파이 배열 차원 확인하기
    print(arr1.shape)
    print(arr2.shape)
(4, 5)
(3, 10)
```

## 배열의 사칙연산

```
## 넘파이 배열 사칙 연산
arr5 = np.array([[1,2,3],
                [4,5,6]
|arr6 = np.array([[7,8,9],
                [10,11,12]])
print("-> arr5+arr6=")
print(arr5+arr6,"\n")
print("->arr5-arr6=")
print(arr5-arr6,"\n")
print("->arr5*arr6=")
print(arr5*arr6,"#n")
print("->arr5/arr6=")
print(arr5/arr6,"\n")
```

```
-> arr5+arr6=
[[ 8 10 12]
 [14 16 18]]
->arr5-arr6=
[[-6 -6 -6]]
 [-6 -6 -6]]
->arr5*arr6=
[[ 7 16 27]
 [40 55 72]]
->arr5/arr6=
[[0.14285714 0.25 0.33333333]
 [0.4 0.45454545 0.5
```