

2025-04-14(월))

crawling

DataScience-master.zip 문서 참고

- 가상환경
 - [2.2 파이썬 가상환경](#) ◆ [wikidocs](#)
 - [1.3. Python 가상환경](#) ◆ [wikidocs](#)
 - [파이썬 가상환경 생성, 삭제, 복사, 옮기기](#)
 - 가상환경 조회 : 가상환경 리스트가 나오고 풀더 명 앞에 * 가 표시되는 곳이 현재 파이썬 환경을 나타냄
 - cmd >> conda info --envs conda info -e
 - 가상환경 실행
 - cmd >> activate 가상환경명칭
 - [한곳에서 끝내는 파이썬 & 머신러닝 & 딥러닝](#)
 - [conda에서 파이썬 가상 환경 \(virtual environments\) 생성하기](#)
 - conda create -n <new_환경명> python=<버전(ex:3.5이나 3.7 등)>
 - 가상환경 생성 : python 3.7버전과 아나콘다 패키지를 같이 설치하여 가상환경 만들
 - conda create --name 가상환경명칭 python=파이썬버전 [같이 설치할 패키지명]
 - cmd >> conda create --name test_project python=3.9 anaconda
 - 가상환경에 추가 패키지 설치 : 가상환경을 activate 시키고 conda install [패키지명] 과 같은 효과임
 - conda install -n 가상환경명 [패키지명]
 - cmd >> conda install -n test_project numpy
 - 가상환경 해제
 - cmd >> deactivate
 - 가상환경 깨끗이 삭제 : --all 옵션을 주어 모든 패키지까지 싹 삭제함
 - conda remove -n 가상환경명 --all
 - cmd >> conda remove -n test_project --all
 - 가상환경 복사 : test_project 가상환경을 복사하여 test_project_v2 로 생성함
 - conda create -n 생성할가상환경명 --clone 원본가상환경명
 - cmd >> conda create -n test_project --clone test_project_v2
 - conda env export -n 가상환경명칭 > 파일명.yml
 - cmd >> conda env export -n test_project > test_project .yml
 - 환경파일로 가상환경을 생성
 - conda env create -f 파일명.yml
 - cmd >> conda env create -f test_project .yml
- ★★ [파이썬으로 위도,경도 찾기\(geocoder, geocoding API, 구글 스프레드시트\)](#)
 1. 구글 스프레드 시트 geocode로 변환
 2. 파이썬 geopy 라이브러리
 3. 구글 geocoding API
- seaborn

- [seaborn](#)
- [User guide and tutorial](#)
- [Seaborn을 사용한 데이터 분포 시각화](#)
- Matplotlib
 - [★ Matplotlib: Visualization with Python](#)
 - [Pyplot tutorial](#)
- Numpy
 - [NumPy](#)
 - [★ 넘파이\(Numpy\) 튜토리얼](#)
 - [★ NumPy fundamentals](#)
- Pandas
 - [pandas documentation](#)
 - [★ API reference](#)
 - [데이터프레임 함수적용 pandas apply 사용법 및 apply lambda 설명](#)

- [★ 교재: Python for Data Analysis, 3E \(Wes's Blog\)](#)
 - [src code - wesm / pydata-book](#)
- Python Code convention
 - [★ PEP 8 – Style Guide for Python Code](#)
 - [\[Python\] 보편적인 python coding convention\(파이썬 코딩 컨벤션\)](#)
 - [Python Code convention Guide](#)
 - [Yosseulsin-JOB/Google-Python-Style-Guide-kor](#)
 - [0.3 파이썬 스타일 가이드 \(PEP-8\) wikidocs](#)
- [★ ★ 파이썬 패키지 인덱스\(PyPI\) : 파이썬 패키지 관리 시스템](#)
 - [★ Q_01_08. PIP\(Python Package Index\) 란 무엇인가 \(wikidocs\)](#)
 - [Q_01_09. PIP 명령문 사용하기](#)
 - [★ tqdm](#)
 - [tqdm derives from the Arabic word taqaddum \(تَقَدُّم\) which can mean “progress,” and is an abbreviation for “I love you so much” in Spanish \(te quiero demasiado\).](#)
 - [Instantly make your loops show a smart progress meter - just wrap any iterable with tqdm\(iterable\), and you're done!](#)
 - [geopy](#)
 - [geopy is a Python client for several popular geocoding web services.](#)

```
>>> from geopy.geocoders import Nominatim
>>> geolocator =
Nominatim(user_agent="specify_your_app_name_here")
>>> location = geolocator.geocode("175 5th Avenue NYC")
>>> print(location.address)
```

```
Flatiron Building, 175, 5th Avenue, Flatiron, New York, NYC,
New York, ...
>>> print((location.latitude, location.longitude))
(40.7410861, -73.9896297241625)
>>> print(location.raw)
{'place_id': '9167009604', 'type': 'attraction', ...}
```

- [googlemaps](#)
 - Python Client for Google Maps Services
- [folium](#)
 - Make beautiful maps with Leaflet.js & Python
 - [folium](#)
- [seaborn](#)
 - Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.
- [requests](#)
 - Requests is a simple, yet elegant, HTTP library.
 - [API Reference and User Guide](#)
- [urllib3](#)
 - a powerful, user-friendly HTTP client for Python

crawling

conda 가상환경 생성 & 활성화

1. conda create crawling
 - conda create -n crawling python=3.12.7
 - conda env remove --name {가상환경명}
2. conda activate crawling
3. mkdir crawling
4. cd crawling
5. . code
6. 가상환경 외부로 이관시
 - 나의 의존성을 파일로 생성
 - pip freeze -> requirements.txt
 - 파일을 읽어서 가상환경 설정
 - pip install -r requirements.txt
7. conda cheatsheet 참고

1. 구글 스프레드 시트 geocode로 변환
2. 파이썬 geopy 라이브러리 ★
 1. 라이브러리 설치
 - !pip install geopy
 2. import 라이브러리
 - from geopy.geocoders import Nominatim
3. 위도, 경도 반환하는 함수

```
def geocoding(address):
    try:
        geo_local = Nominatim(user_agent='South Korea') #지역설정
        location = geo_local.geocode(address)
        geo = [location.latitude, location.longitude]
        return geo

    except:
        return [0,0]

# 실행
for idx,addr in enumerate(tqdm(df_shake.addr)):
    df_shake.loc[idx,'latitude'] = geocoding(addr)[0]
    df_shake.loc[idx,'longitude'] = geocoding(addr)[1]
```

3. 구글 geocoding API

[pandas.DataFrame.pivot_table](#)

- The levels in the pivot table will be stored in MultiIndex objects (hierarchical indexes) on the index and columns of the result DataFrame.

```
DataFrame.pivot_table(values=None, index=None, columns=None,
aggfunc='mean', fill_value=None, margins=False, dropna=True,
margins_name='All', observed=<no_default>, sort=True)[source]
```