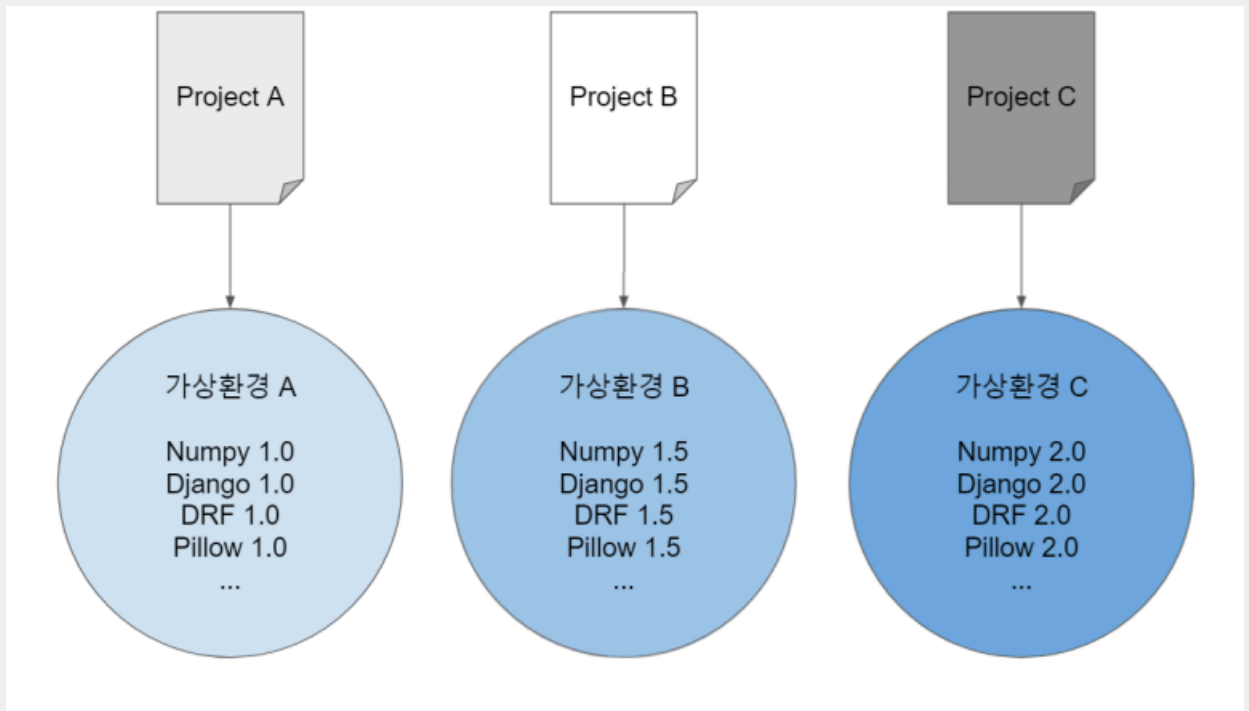


# 가상환경 만들기

아나콘다는 자신만의 가상환경을 이미 가지고 있습니다. 프롬프트 앞에 (base) 라고 추가되어 있는것이 바로 가상환경의 이름입니다.

## 가상 환경(Virtual Environments)이란?

- 가상 환경은 독립적인 파이썬 실행 환경을 의미합니다.
- 각각의 가상 환경은 다른 환경에 영향을 미치지 않으며 각자 다른 종류와 버전의 패키지를 가질 수 있습니다.
- 만약 프로젝트를 여러개 진행하는 경우 다른 버전의 패키지와 호환이 되지 않아 문제가 발생하는 경우가 있습니다.
- 이런 경우 가상환경에서 진행한다면 Project A 가 B 에 전혀 영향을 미치지 않으므로 호환이 되지 않는 문제를 미연에 방지할 수 있습니다.



우리만의 가상환경을 만들어 보겠습니다.

다음은 'pydatavenv' 라는 이름으로 가상환경을 만들기 위한 명령입니다.

```
(base) C:\Users\user>conda create --name pydatavenv python=3.8
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
current version: 4.10.1
```

latest version: 4.10.3

Please update conda by running

```
$ conda update -n base -c defaults conda
```

.....블라블라

Proceed ([y]/n)? y

```
Anaconda Prompt (anaconda3) - conda create --name pydatavenv python=3.8

Total: 22.7 MB

The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2021.9.30-haa95532_1
certifi              pkgs/main/win-64::certifi-2021.10.8-py38haa95532_0
openssl              pkgs/main/win-64::openssl-1.1.1l-h2bbff1b_0
pip                  pkgs/main/win-64::pip-21.0.1-py38haa95532_0
python               pkgs/main/win-64::python-3.8.12-h6244533_0
setuptools           pkgs/main/win-64::setuptools-58.0.4-py38haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.36.0-h2bbff1b_0
vc                   pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                pkgs/main/noarch::wheel-0.37.0-pyhd3eb1b0_1
wincertstore         pkgs/main/win-64::wincertstore-0.2-py38haa95532_2

Proceed ([y]/n)? y

Downloading and Extracting Packages
wincertstore-0.2      | 15 KB      | ##### | 100%
openssl-1.1.1l       | 4.8 MB     | ##### | 100%
wheel-0.37.0         | 33 KB      | ##### | 100%
sqlite-3.36.0        | 780 KB     | ##### | 100%
ca-certificates-2021 | 116 KB     | ##### | 100%
python-3.8.12        | 16.0 MB    | #####7 | 98%
```

가상환경이 만들어지면 다음 명령을 통해 확인해 봅니다.

(base) C:\Users\nakja> conda info --envs

```
Anaconda Prompt (anaconda3)

done
#
# To activate this environment, use
#
#     $ conda activate mydatavenv
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

(base) C:\Users\nakja>conda info --envs
# conda environments:
#
base                * C:\W01DevelopKits\anaconda3
mydatavenv          C:\W01DevelopKits\anaconda3\envs\mydatavenv

(base) C:\Users\nakja>
```

가상환경을 활성화 한 후 jupyter lab의 커널로 등록합니다.

맥북에서 오류날때

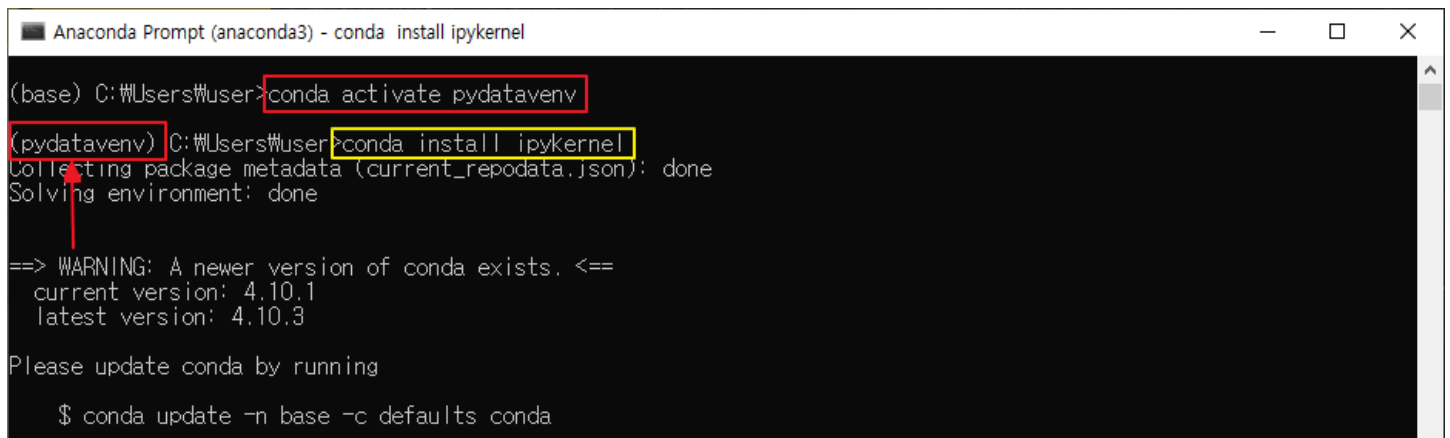
**conda init --all**

**source ~/anaconda3/etc/profile.d/conda.sh**

// 경로 확인해서 나는 **source /opt/anaconda3/etc/profile.d/conda.sh** 임.

```
(base) C:\Users\nakja>conda activate pydatavenv

(mydatavenv) C:\Users\nakja>conda install ipykernel
Collecting package metadata (current_repodata.json): done
Solving environment: done
```



```
Anaconda Prompt (anaconda3) - conda install ipykernel

(base) C:\Users\User>conda activate pydatavenv

(pydatavenv) C:\Users\User>conda install ipykernel
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.1
  latest version: 4.10.3

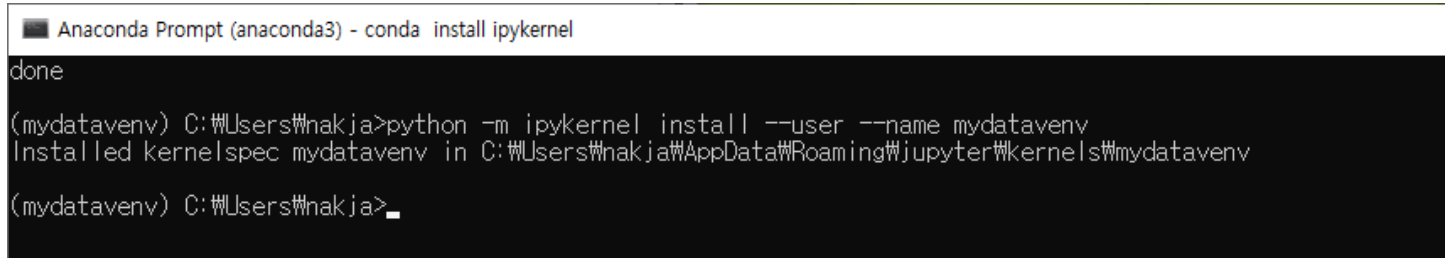
Please update conda by running

  $ conda update -n base -c defaults conda
```

Proceed ([y]/n)? 를 물어보면 y를 입력합니다.

완료되면 우리가 만든 가상환경을 jupyter lab의 커널로 등록합니다.

```
(pydatavenv) C:\Users\nakja>python -m ipykernel install --user --name pydatavenv
Installed kernelspec pydatavenv in C:\Users\user\AppData\Roaming\jupyter\kernels\pydatavenv
```



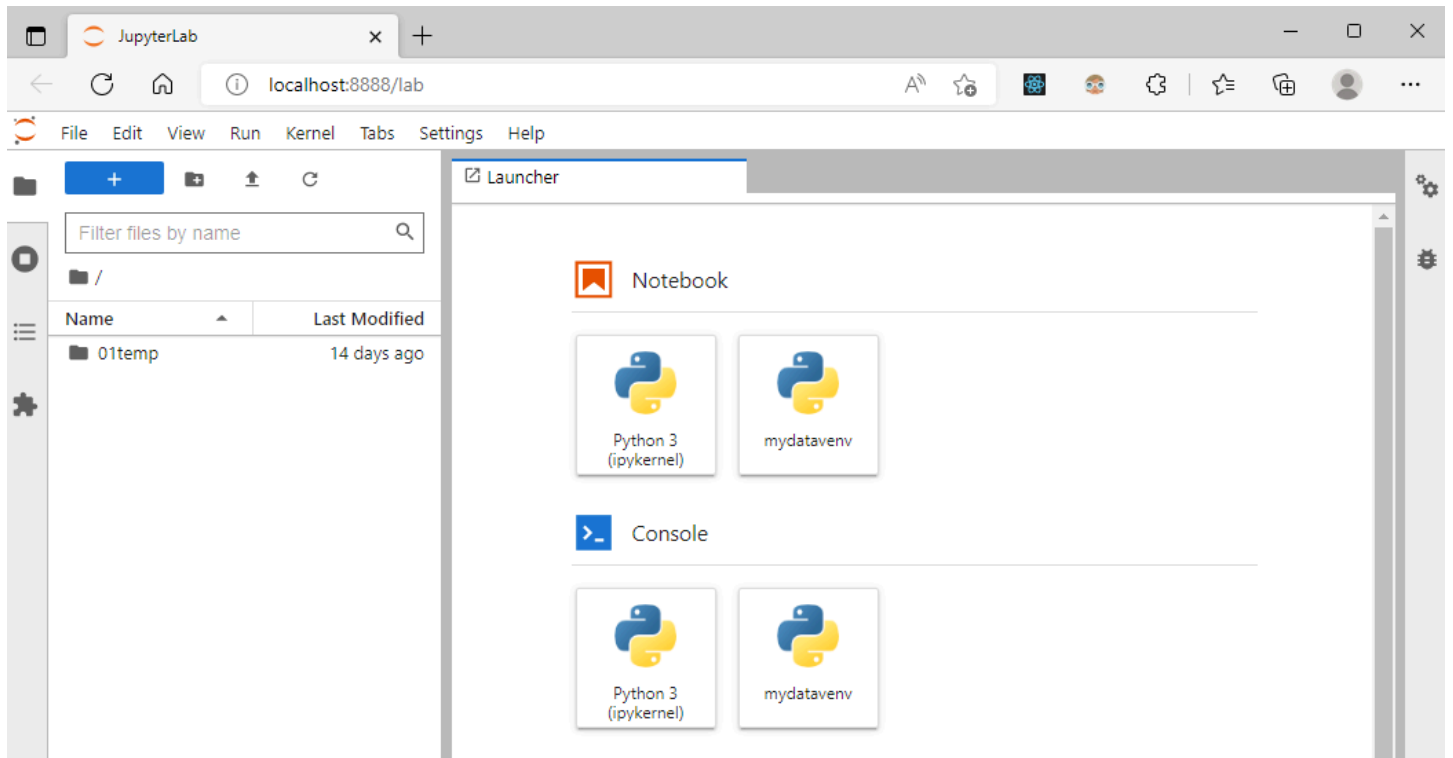
```
Anaconda Prompt (anaconda3) - conda install ipykernel

done

(mydatavenv) C:\Users\nakja>python -m ipykernel install --user --name mydatavenv
Installed kernelspec mydatavenv in C:\Users\nakja\AppData\Roaming\jupyter\kernels\mydatavenv

(mydatavenv) C:\Users\nakja>_
```

등록이 완료되면 Anaconda Prompt 창을 한개 더 오픈한 후 jupyter lab을 실행합니다.  
그러면 아래와 같이 우리가 추가한 가상환경이 추가된것을 확인할 수 있습니다.



우리가 선택한 워크스페이스에서 가상환경을 jupyter lab을 통해 실행할 수 있도록 설정이 완료되었습니다.

앞에서 생성한 가상환경 내에 필요한 라이브러리를 설치합니다.

아래에 제시된 내용은 반드시 필요한 것은 아니고, 필요한 경우 참조해서 설치하면 됩니다.

pandas 나 matplotlib 등을 설치해 봅니다.

```
(pydatavenv) C:\Users\user>conda install pillow
```

```
Collecting package metadata (current_repodata.json): done
```

```
Solving environment: done
```

```
(pydatavenv) C:\Users\user>conda install pandas
```

```
Collecting package metadata (current_repodata.json): done
```

```
Solving environment: done
```

```
(pydatavenv) C:\Users\user>conda install matplotlib
```

```
Collecting package metadata (current_repodata.json): done
```

```
Solving environment: done
```

```
(pydatavenv) C:\Users\user>conda install seaborn
```

Collecting package metadata (current\_repodata.json): done

Solving environment: done

(pydatavenv) C:\Users\user>pip install folium

Collecting package metadata (current\_repodata.json): done

Solving environment: done

---아래는 당장은 필요없고, 필요시 하나씩 설치하면 된다.

(pydatavenv) C:\Users\user>conda install scikit-learn

Collecting package metadata (current\_repodata.json): done

Solving environment: done

(pydatavenv) C:\Users\user>conda install xlrd

Collecting package metadata (current\_repodata.json): done

Solving environment: done

(pydatavenv) C:\Users\user>conda install -c conda-forge googlemaps

Collecting package metadata (current\_repodata.json): done

Solving environment: done

(pydatavenv) C:\Users\user>conda install openpyxl

Collecting package metadata (current\_repodata.json): done

Solving environment: done