

# 윈도우즈10에 CUDA, cuDNN, Anaconda, Jupyter, Tensorflow 설치, VSCode, Jupyter 연동



dohyoung rim(이)가 만들  
조금 전에 마지막 업데이트 • 🔍 페이지를 조회한 사람 3명

Anacoda는 Python과 기타 중요 라이브러리에 대한 통합 환경이다.

Anaconda를 설치하면 개별적으로 python, jupyter 설치를 하지 않아도 된다.

## 전제

python, jupyter는 설치 되어 있지 않아도 된다.

VSCode와 연동을 위해서는 VSCode가 설치되어 있어야 한다.

## 설치 시나리오

GPU 드라이버 설치

CUDA 설치

cuDNN 설치

Anaconda를 설치하고

가상환경 my\_env를 생성하고

가상환경 my\_env 안에서 tensorflow를 설치.

VSCode에서 my\_env를 연동하고 tensorflow 로딩 확인

jupyter에서 가상환경을 사용하기 위해 jupyter 커널에 my\_env를 등록

jupyter에서 my\_env에 연동하여 노트북 생성하고 tensorflow 로딩 확인

## GPU 드라이버 설치

<https://www.nvidia.com/download/index.aspx?lang=kr> 페이지에서 다음 항목을 설정 하고 '검색' 클릭.

### NVIDIA 드라이버 다운로드

아래의 드롭다운 목록에서 자신의 NVIDIA 제품에 알맞은 드라이버를 선택하세요. 1

제품 유형:	GeForce
제품 시리즈:	GeForce 16 Series
제품 계열:	GeForce GTX 1660 Ti
운영 체제:	Windows 10 64-bit
다운로드 타입:	Game Ready 드라이버(GRD)
언어:	Korean

검색

'다운로드' 클릭.

## GEFORCE GAME READY

버전: 471.11 WHQL.  
배포 날짜: 2021.6.22  
운영 체제: Windows 10 64-bit  
언어: Korean  
파일 크기: 720 MB

다운로드

다운로드 클릭.

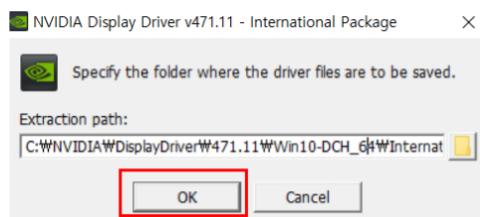
## 다운로드

이 다운로드에는 NVIDIA 그래픽 드라이버와 GeForce Experience 애플리케이션을 추가로 설치하는 옵션이 포함되어 있습니다. 소프트웨어 사용에 대한 자세한 내용은 [NVIDIA GeForce Software 라이센스](#)와 [GeForce Experience Software 라이센스](#)를 각각 참조하세요.

다운로드

다운로드 받은 xxx.xx-desktop-win10-64bit-xxxx.exe를 실행.

실행된 창에서 'OK'를 클릭.

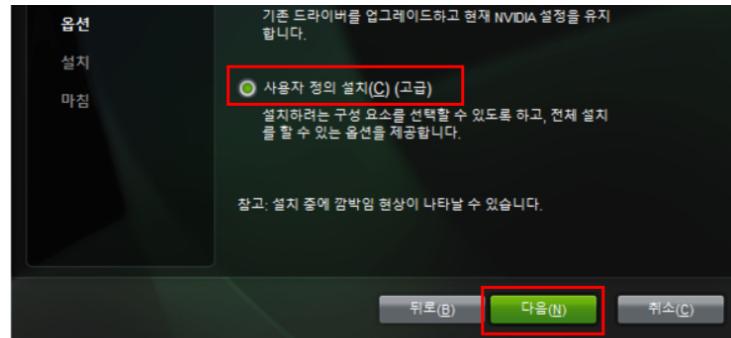


'NVIDIA 그래픽 드라이버'를 선택하고 '동의 및 계속' 클릭.

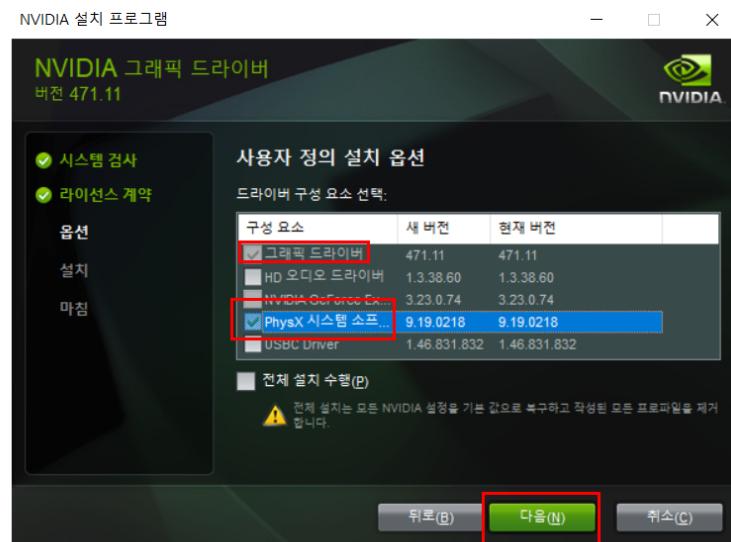


'사용자 정의 설치' 선택하고 '다음' 클릭





'그래픽 드라이버'와 'PhysX 시스템 소프트웨어'만 체크하고 '다음' 클릭.



## CUDA 설치

<https://developer.nvidia.com/cuda-toolkit-archive> 페이지에서 CUDA Toolkit 11.4.0 을 클릭.



Previous releases of the CUDA Toolkit, GPU Computing SDK, documentation and tools are listed below, and be sure to check [www.nvidia.com/drivers](http://www.nvidia.com/drivers) for more recent products.

[Download Latest CUDA Toolkit](#)

[Latest Release](#)

[CUDA Toolkit 11.4.0 \[June 2021\], Versioned Online Documentation](#)

[Archived Releases](#)

[CUDA Toolkit 11.3.1 \[May 2021\], Versioned Online Documentation](#)

[CUDA Toolkit 11.3.0 \[April 2021\], Versioned Online Documentation](#)

[CUDA Toolkit 11.2.2 \[March 2021\], Versioned Online Documentation](#)

[CUDA Toolkit 11.2.1 \[Feb 2021\], Versioned Online Documentation](#)

[CUDA Toolkit 11.2.0 \[Dec 2020\], Versioned Online Documentation](#)

'Windows', 'x86\_64', '10', 'exe(local)'을 클릭.

## Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be listed. See the terms and conditions of the [CUDA EULA](#).

Operating System

Linux

Windows

Architecture

x86\_64

Version

10

Server 2016

Server 2019

Installer Type

exe (local)

exe (network)

'Download(...)' 클릭.

## Download Installer for Windows 10 x86\_64

The base installer is available for download below.

►Base Installer

Download (2.8 GB)

Installation Instructions:

1. Double click cuda\_11.4.0\_471.11\_win10.exe
2. Follow on-screen prompts

The checksums for the installer and patches can be found in [Installer Checksums](#).

For further information, see the [Installation Guide for Microsoft Windows](#) and the [CUDA Quick Start Guide](#).

다운로드 받은 cuda\_11.xxx.exe를 더블 클릭.

압출풀 위치를 정하고 OK 클릭. 이미 있는 폴더는 에러가 난다. 패스 맨뒤에 tmp를 붙여준다.



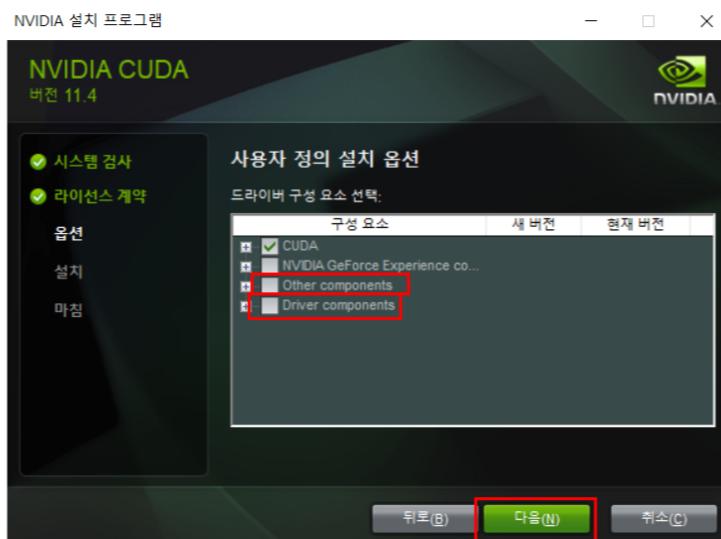
압축을 푼 폴더에서 setup.exe를 더블 클릭.

PPC	2021-07-19 오후 12:59	파일 폴더
ShadowPlay	2021-07-19 오후 12:59	파일 폴더
ShieldWirelessController	2021-07-19 오후 12:59	파일 폴더
universal	2021-07-19 오후 12:58	파일 폴더
Update.Core	2021-07-19 오후 12:59	파일 폴더
visual_studio_integration	2021-07-19 오후 12:58	파일 폴더
Colinstaller32.exe	2021-05-18 오후 5:25	응용 프로그램
Colinstaller64.exe	2021-05-18 오후 5:25	응용 프로그램
EULA.txt	2021-05-28 오전 11:17	텍스트 문서
license.txt	2021-05-18 오후 5:25	텍스트 문서
ListDevices.txt	2021-06-21 오후 5:34	텍스트 문서
NVMUP.cfg	2021-05-18 오후 5:25	CFG 파일
setup.cfg	2021-06-24 오전 11:39	CFG 파일
setup.exe	2021-05-18 오후 5:25	응용 프로그램

'사용자 정의 설치' 선택, '동의 및 계속' 클릭.



'CUDA'만 체크하고 나머지는 체크를 푼다. '다음' 클릭.

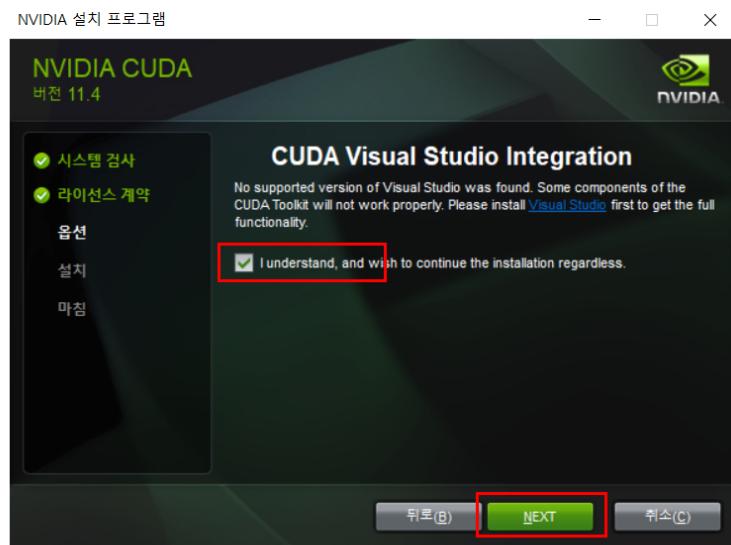


'다음' 클릭.

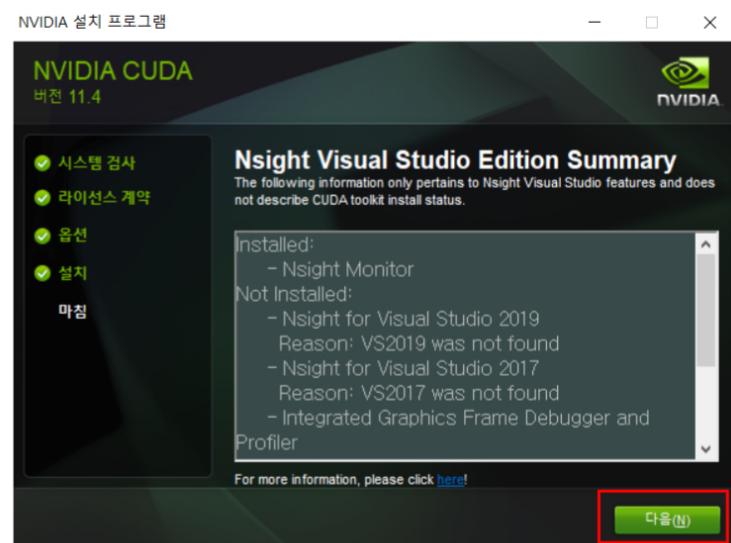




'I understand, ...' 체크하고 'NEXT' 클릭.



'다음' 클릭.



'닫기' 클릭.





아나콘다 파워셸을 실행시키고 다음을 실행하여 잘 설치된 것을 확인한다.

```
1 nvcc --version
```

```
■ Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\DMC CONET> nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Wed_Jun_2_19:25:35_Pacific_Daylight_Time_2021
Cuda compilation tools, release 11.4, V11.4.48
Build cuda_11.4.r11.4/compiler.30033411_0
(base) PS C:\Users\DMC CONET>
```

## cuDNN 설치

<https://developer.nvidia.com/cudnn>에서 'Download cuDNN' 클릭.

## NVIDIA cuDNN

The NVIDIA CUDA® Deep Neural Network library (cuDNN) is a GPU-accelerated library of primitives for deep neural networks. It provides highly optimized, fast, and portable implementations for standard routines such as forward and backward convolution, pooling, matrix multiplication, and more.

Deep learning researchers and framework developers worldwide rely on cuDNN for high performance and portability. By providing optimized primitives for common operations in deep learning frameworks, cuDNN enables researchers and developers to quickly build and train deep learning networks and developing software applications rather than spending time on low-level optimizations. For more information about cuDNN, visit the [NVIDIA GPU CLOUD](#).

Download cuDNN > GTC2020 >

이후 로그인과 설문을 진행.

Agree에 체크하고 cuDNN v8.2.2 for CUDA 11.4를 클릭.

# cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

Agree To the Terms of the cuDNN Software License Agreement

Note: Please refer to the [Installation Guide](#) for release prerequisites, including support

For more information, refer to the cuDNN Developer Guide, Installation Guide and Rele

[Download cuDNN v8.2.2 \[July 6th, 2021\], for CUDA 11.4](#)

[Download cuDNN v8.2.2 \[July 6th, 2021\], for CUDA 10.2](#)

Archived cuDNN Releases

cuDNN Library for Windows (x86)을 클릭.

## Library for Windows and Linux, Ubuntu(x86\_64, arm64)

[cuDNN Library for Linux \(aarch64\)](#)

[cuDNN Library for Linux \(x86\\_64\)](#)

[cuDNN Library for Linux \(PPC\)](#)

[cuDNN Library for Windows \(x86\)](#)

[cuDNN Runtime Library for Ubuntu20.04 x86\\_64 \[Deb\]](#)

[cuDNN Developer Library for Ubuntu20.04 x86\\_64 \[Deb\]](#)

[cuDNN Code Samples and User Guide for Ubuntu20.04 x86\\_64 \[Deb\]](#)

[cuDNN Runtime Library for Ubuntu20.04 aarch64 \[Deb\]](#)

[cuDNN Developer Library for Ubuntu20.04 aarch64 \[Deb\]](#)

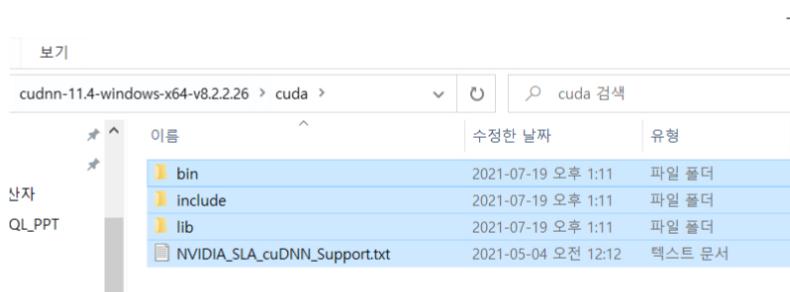
[cuDNN Code Samples and User Guide for Ubuntu20.04 aarch64 \[Deb\]](#)

[cuDNN Cross-compile Library for Ubuntu20.04 aarch64 \[Deb\]](#)

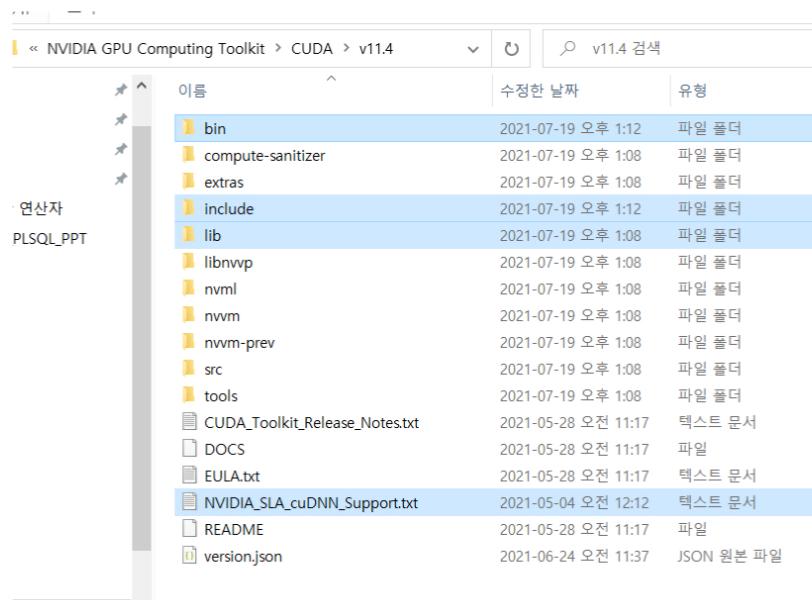
[cuDNN Developer Cross-compile Library for Ubuntu20.04 aarch64 \[Deb\]](#)

다운로드 받은 cudnn-11.4-windows-x64-xxxx.zip 파일의 압축을 푼다.

cuda 폴더안의 파일과 폴더를 모두 복사하여

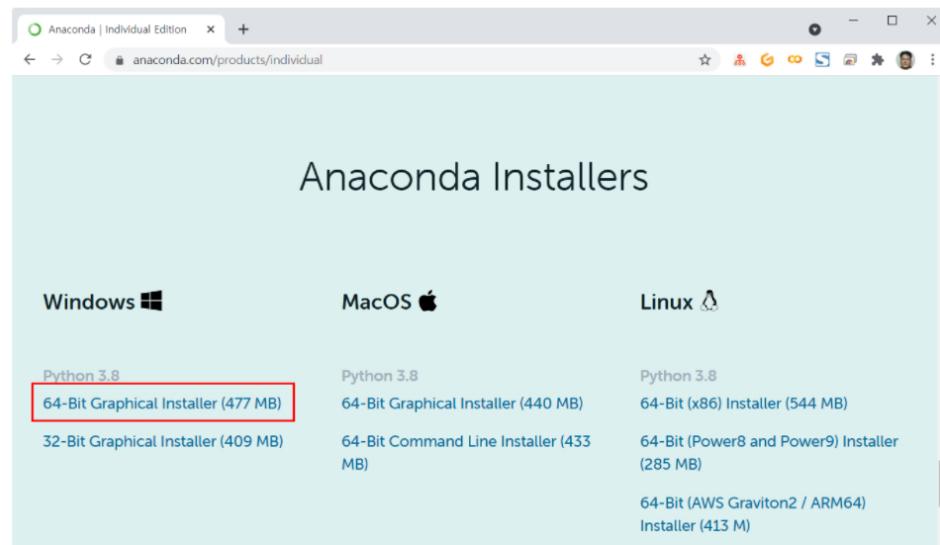


C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.4 폴더안에 붙여넣기 한다.

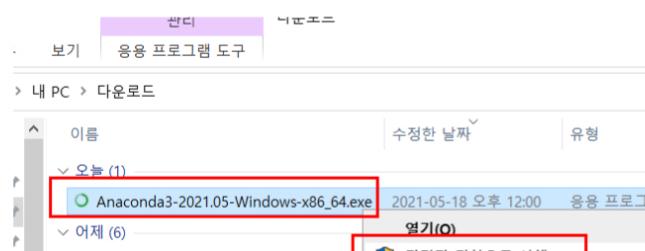


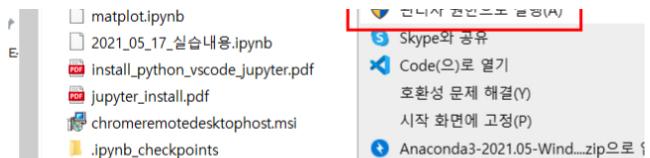
## Anaconda 설치

<https://www.anaconda.com/products/individual>에서 Windows 밑의 64-bit xxx Installer를 클릭.

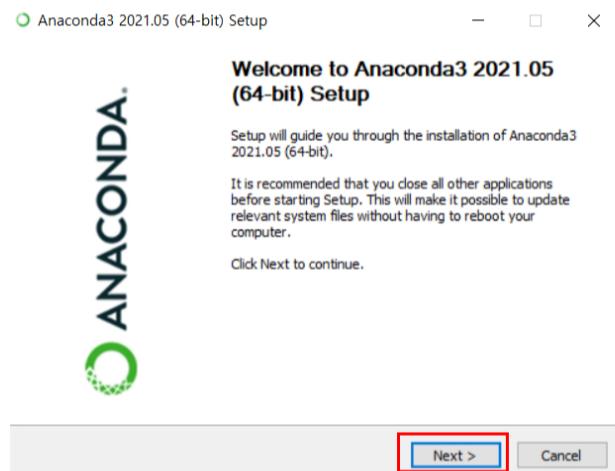


다운로드 받은 Anaconda3-xxxx-Windows-x86\_64.exe에 우측클릭하고 '관리자 권한으로 실행'을 클릭.

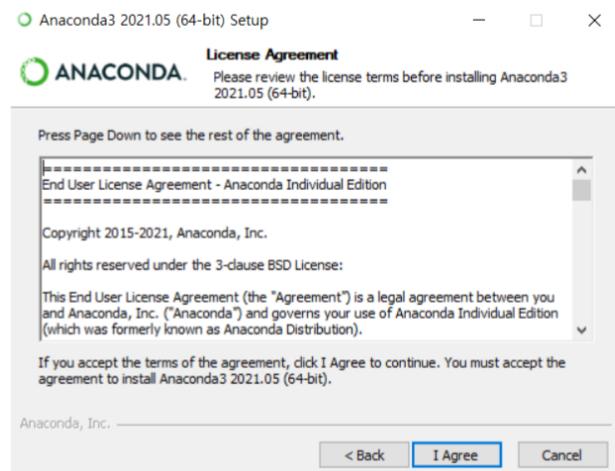




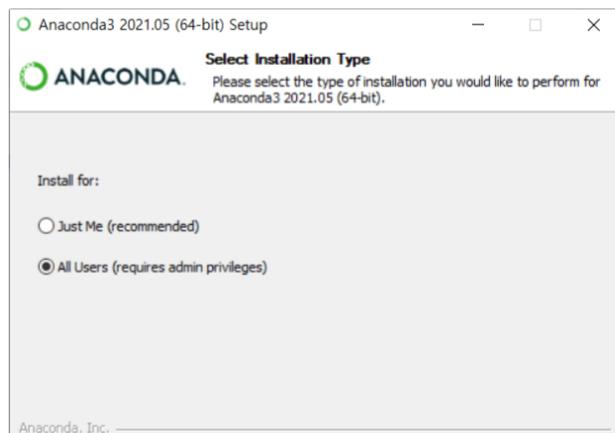
'Next' 클릭.



'I Agree' 클릭.

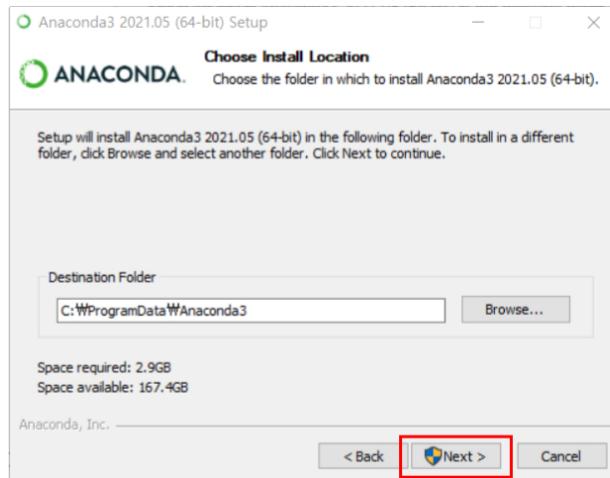


'Next' 클릭.

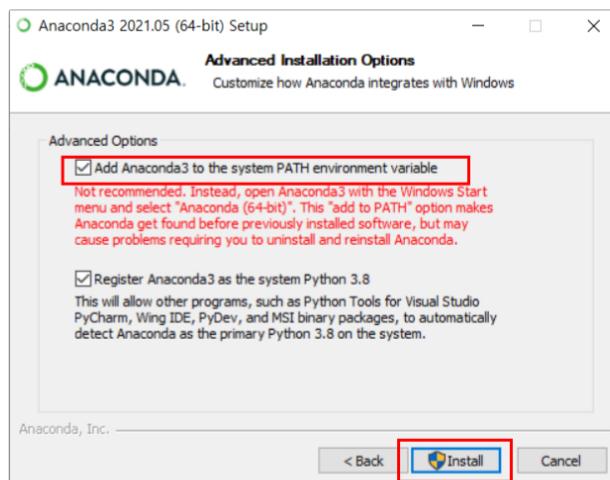


< Back **Next >** Cancel

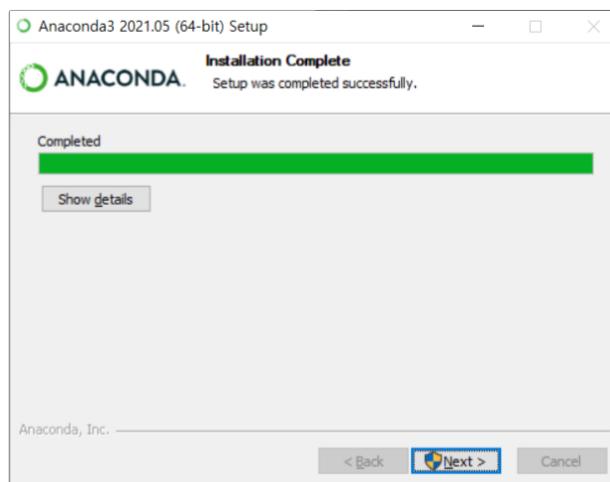
'Next' 클릭.



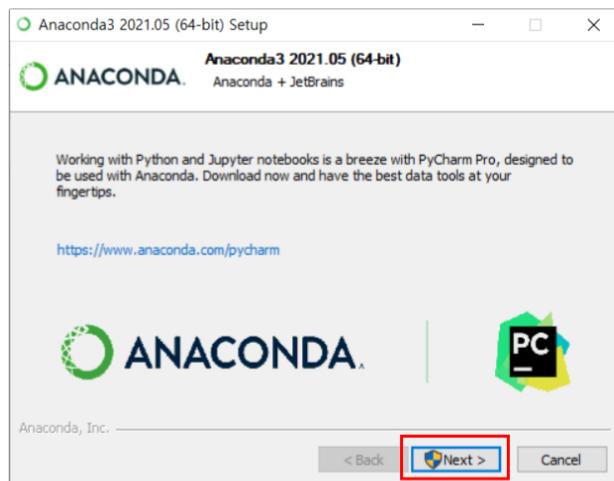
'Add Anaconda3 to the system PATH environment variable'을 체크하고 'Install' 클릭.



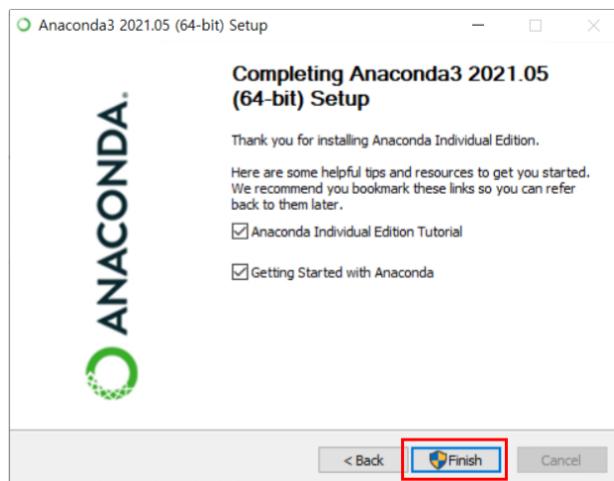
'Next' 클릭.



'Next' 클릭.

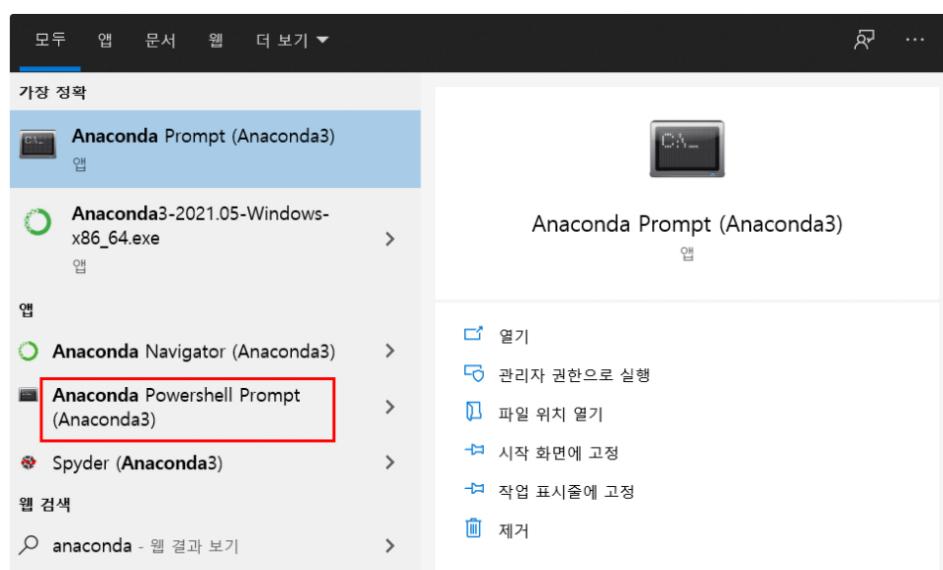


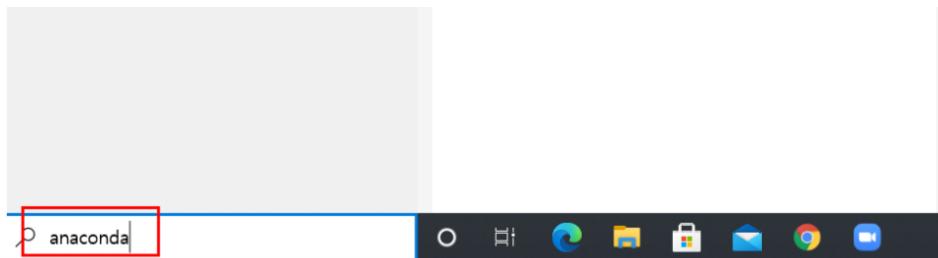
'Finish' 클릭.



## Anaconda 설치 확인

원도우 실행 창에 'anaconda'를 입력하고 'anaconda Powershell Prompt'를 실행.





실행된 창에서 다음을 입력

```
1 python
```

```
■ Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\DMC CONET> python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> -
```

python 프롬프트에서 다음을 입력하여 설치된 라이브러리 임포팅되는지 확인.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
```

```
■ Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\DMC CONET> python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas as pd
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> -
```

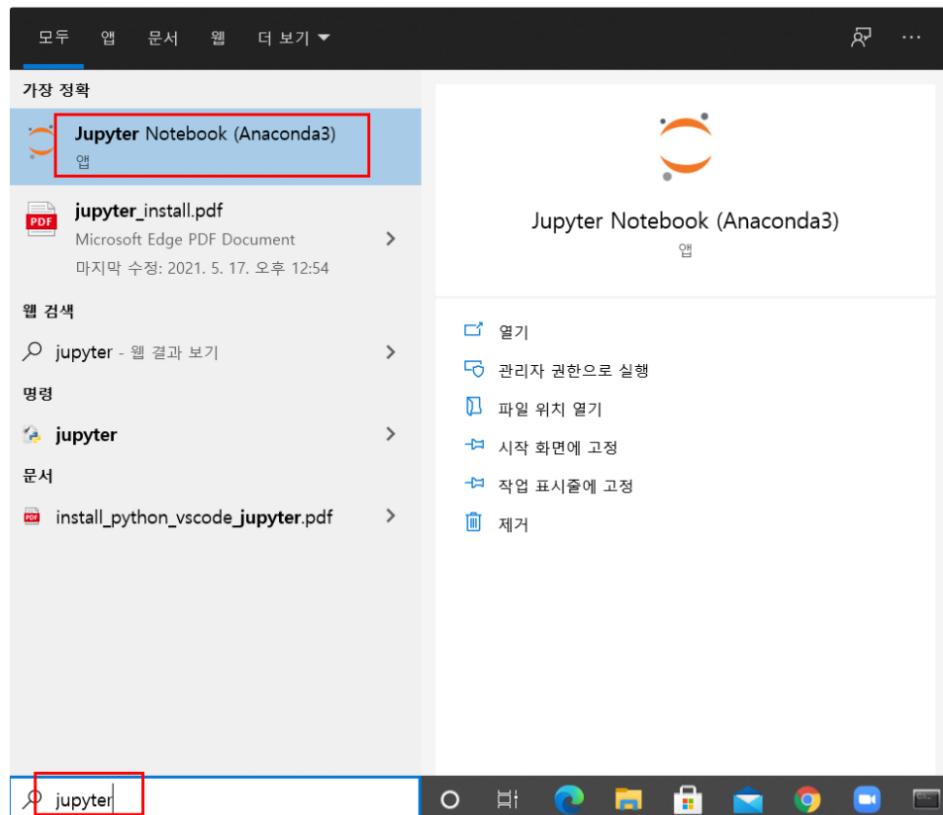
에러 메시지가 보이지 않으면 설치되어 있는 것이고 잘 임포팅 된 것이다.

다음을 입력하여 python 종료.

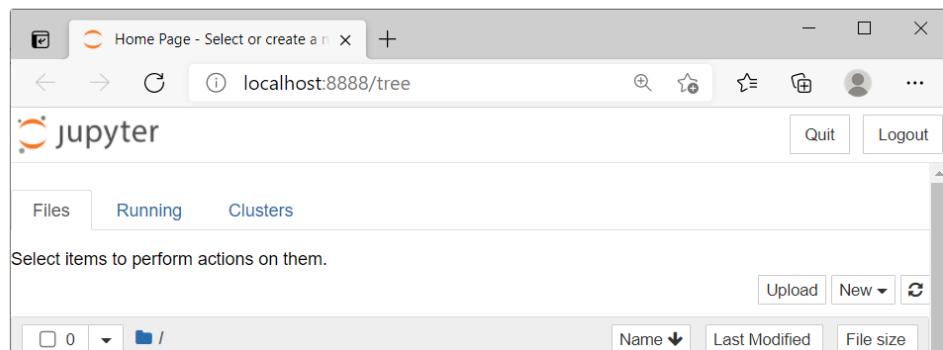
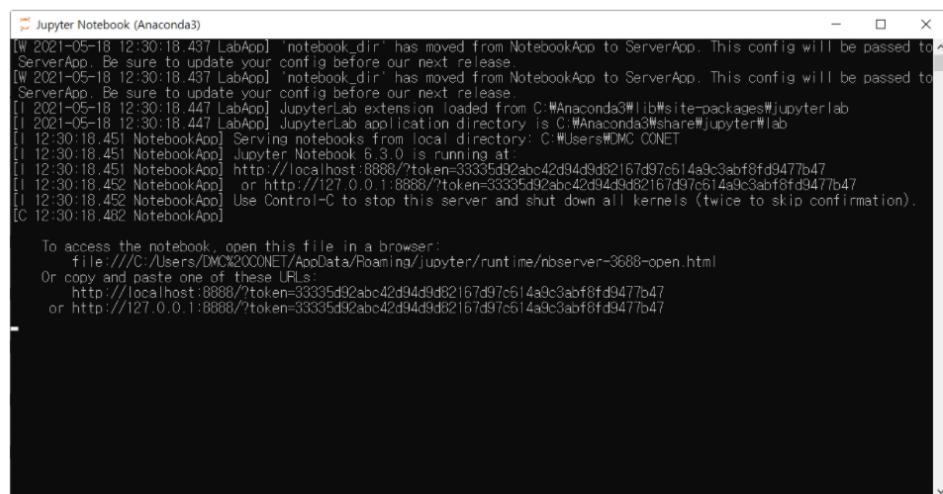
```
1 exit()
```

```
■ Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\DMC CONET> python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916
Type "help", "copyright", "credits" or "license" for more
>>> import pandas as pd
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> exit()
(base) PS C:\Users\DMC CONET>
```

원도우 실행창에 'jupyter'를 입력하고 'Jupyter notebook (Anaconda3)'를 실행.

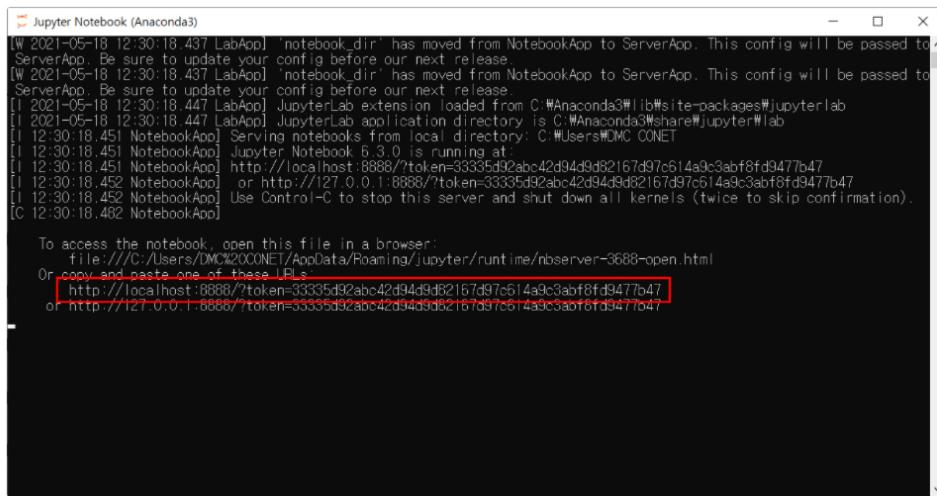


웹 창이 뜨고 곧이어 웹브라우어가 뜨면서 jupyter화면이 보인다.



□	3D Objects	7일 전
□	Contacts	7일 전
□	Desktop	7일 전
□	Documents	7일 전
□	Downloads	31분 전
□	Favorites	7일 전
□	Links	7일 전

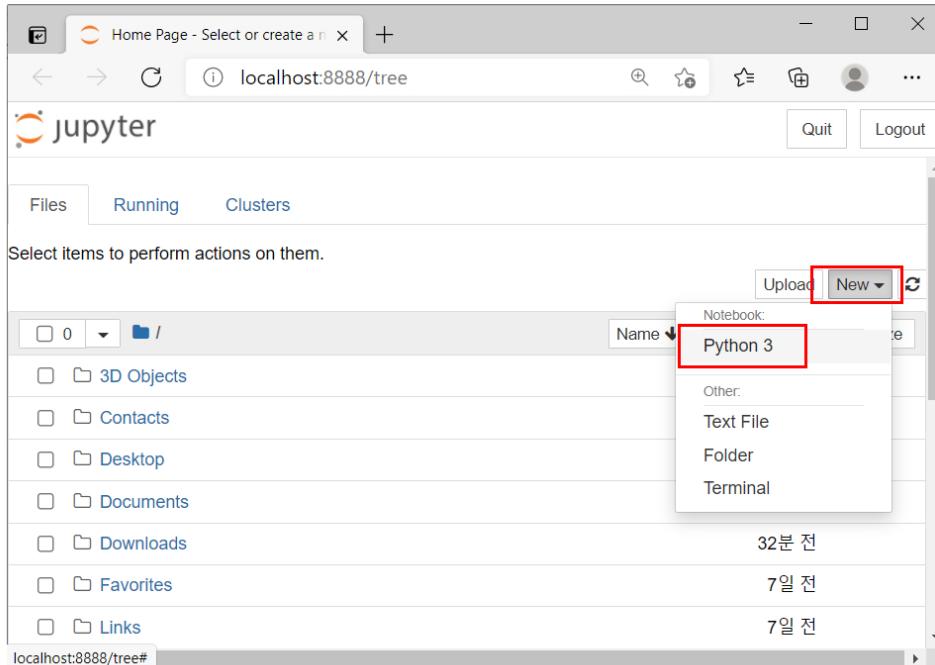
만약 웹브라우저가 뜨지 않고 jupyter 화면이 보이지 않으면, 실행된 셸창에 출력된 다음 url을 카피하여 웹브라우저를 실행하고 주소창에 url을 입력하여 이동한다.



```
[W 2021-05-18 12:30:18.437 LabApp] notebook_dir has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2021-05-18 12:30:18.437 LabApp] notebook_dir has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2021-05-18 12:30:18.447 LabApp] JupyterLab extension loaded from C:\Anaconda3\lib\site-packages\jupyterlab
[I 2021-05-18 12:30:18.447 LabApp] JupyterLab application directory is C:\Anaconda3\share\jupyter\lab
[I 12:30:18.451 NotebookApp] Serving notebooks from local directory: C:\Users\DMC\OneDrive\...
[I 12:30:18.451 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[I 12:30:18.451 NotebookApp] http://localhost:8888/?token=33335d92abc42d94d9d2167d97c614a9c3abf8fd9477b47
[I 12:30:18.452 NotebookApp] or http://127.0.0.1:8888/?token=33335d92abc42d94d9d2167d97c614a9c3abf8fd9477b47
[C 12:30:18.482 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).

To access the notebook, open this file in a browser:
  file:///C:/Users/DMC/OneDrive/AppData/Roaming/jupyter/runtime/nbserver-3688-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=33335d92abc42d94d9d2167d97c614a9c3abf8fd9477b47
  or http://127.0.0.1:8888/?token=33335d92abc42d94d9d2167d97c614a9c3abf8fd9477b47
```

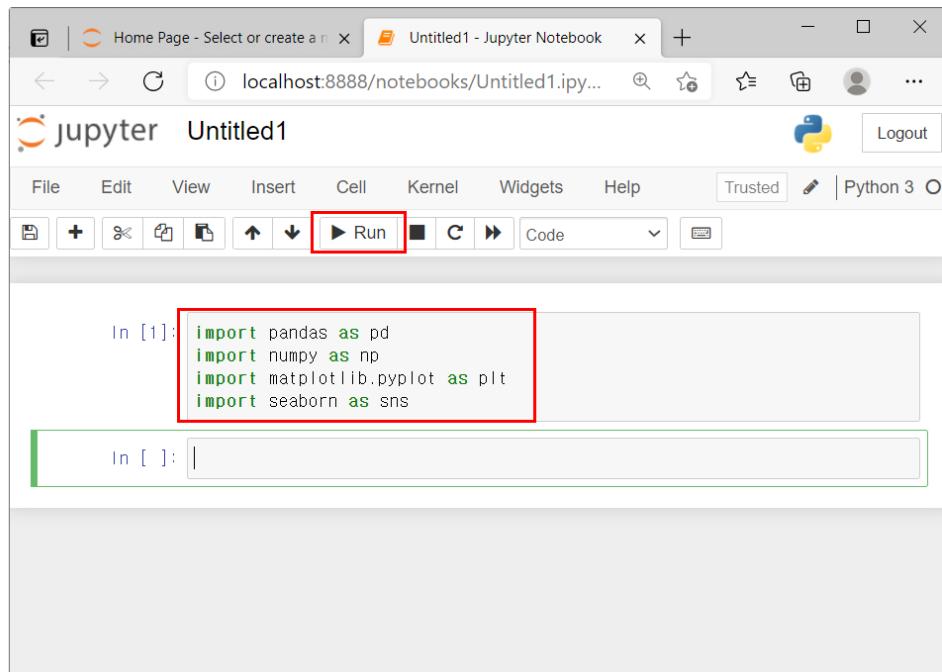
Jupyter 화면에서 'New' > 'Python 3' 클릭.



새로 열린 노트북에 다음을 입력하고 실행 시켜 에러 메시지가 안 뜨는지 확인.

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
```

```
3 import matplotlib.pyplot as plt  
4 import seaborn as sns
```



## 가상환경 생성과 사용

'Anaconda Powershell Prompt (Anaconda3)'을 실행.

**conda env list**

현재 생성된 가상환경 리스트 출력

```
(base) PS C:\Users\DMC CONET> conda env list  
# conda environments:  
#  
base * C:\Anaconda3  
(base) PS C:\Users\DMC CONET>
```

**conda create -n my\_env**

'my\_env'라는 가상 환경 생성

```
(base) PS C:\Users\DMC CONET> conda create -n my_env  
Collecting package metadata (current_repodata.json): done  
Solving environment: done  
## Package Plan ##  
environment location: C:\Anaconda3\envs\my_env  
  
Proceed ([y]/n)? y  
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done  
#  
# To activate this environment, use  
#  
#     $ conda activate my_env  
#  
# To deactivate an active environment, use  
#  
#     $ conda deactivate
```

```
(base) PS C:\Users\DMC CONET>
```

다시 'conda env list'를 하면 환경 리스트에 my\_env 이름으로 추가되었다.

```
(base) PS C:\Users\DMC CONET> conda env list
# conda environments:
#
base                         * C:\Anaconda3
my_env                         C:\Anaconda3\envs\my_env
(base) PS C:\Users\DMC CONET>
```

#### conda activate my\_env

가상환경 my\_env으로 들어가기. 해당 가상환경을 사용하게 된다.

```
(base) PS C:\Users\DMC CONET> conda activate my_env
(my_env) PS C:\Users\DMC CONET>
```

현재 환경의 이름 my\_env가 맨 앞에 보인다.

#### conda deactivate

가상환경에서 나가기.

```
(my_env) PS C:\Users\DMC CONET> conda deactivate
(base) PS C:\Users\DMC CONET>
```

## TensorFlow 설치

가상환경에서 나가고 base에서 실행한다.

```
1 pip install --user tensorflow-gpu
2
3 conda install tensorflow-gpu
```

중간에 'y'를 입력한다.

설치 완료되면 다음과 같다.

```
[■] Anaconda Powershell Prompt (Anaconda3)
done
(my_env) PS C:\Users\DMC CONET>
```

💡 pip install과 conda install 둘다 해주었다. conda install만 하면 웬지 몰라도 tensorflow에서 GPU를 인식하지 못한다.

python 실행 후 다음 실행으로 tensorflow 설치와 임포팅됨을 확인할 수 있다.

```
1 import tensorflow as tf
2 print(tf.__version__)
```

```
[my_env] PS C:\Users\DMC CONET> python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit]
Type "help", "copyright", "credits" or "license" for more information
>>> import tensorflow as tf
>>> print(tf.__version__)
2.3.0
>>>
```

다음 코드 실행으로 tensorflow에서 GPU를 인식함을 확인할 수 있다.

```
1 from tensorflow.python.client import device_lib
2 print(device_lib.list_local_devices())
```

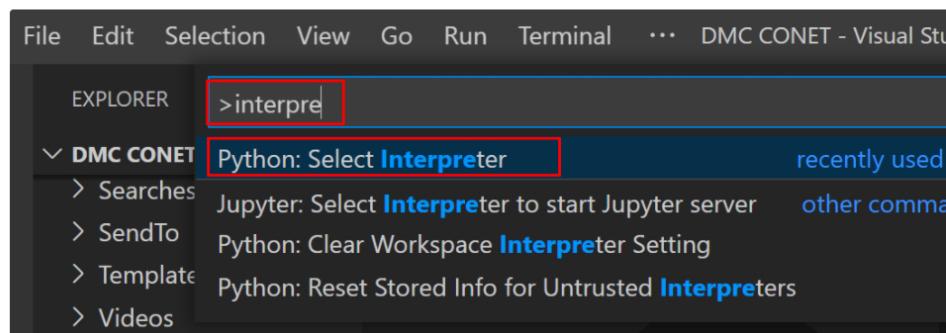
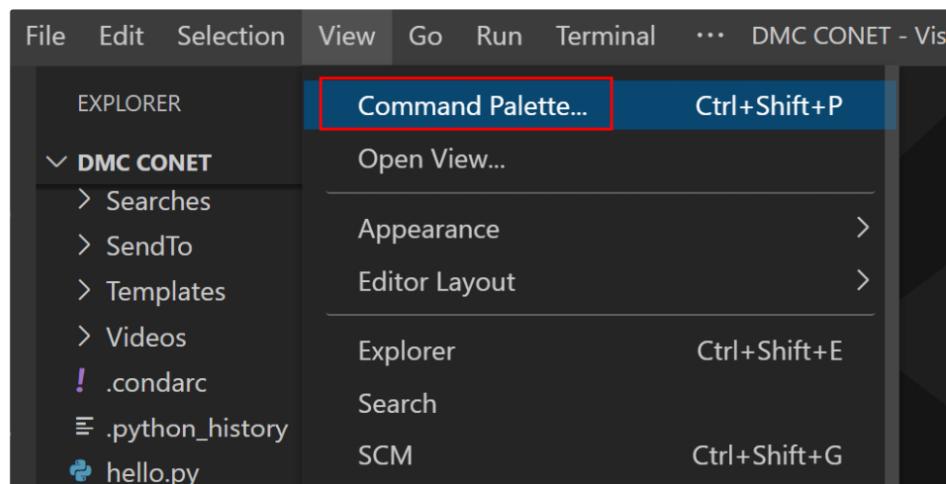
실행 결과

```
1 /device:GPU:0
```

## 가상 환경과 VSCode와 연동

VSCode를 실행.

'View' > 'Command Palette...'를 클릭.

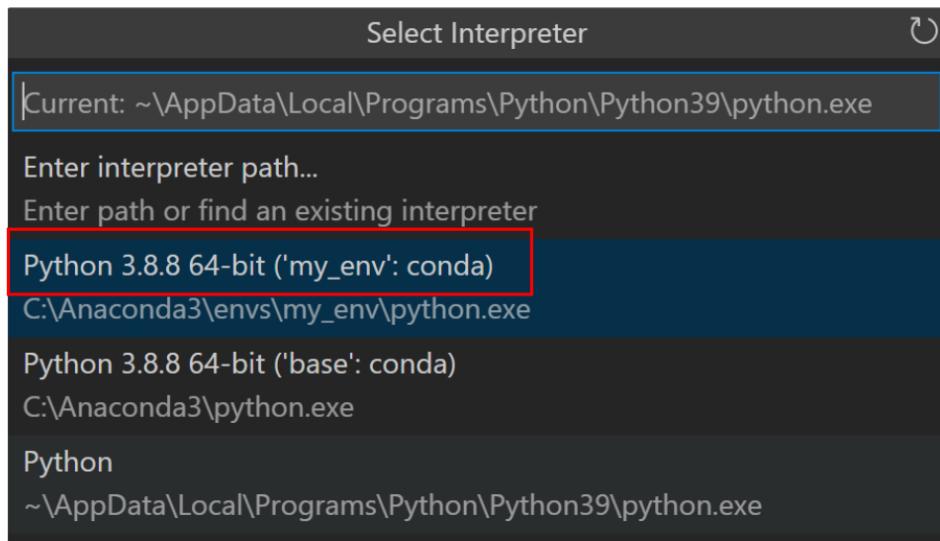




기본 가상환경 'base'와 생성한 가상환경 'my\_env'가 보입니다.

이외에 별개로 설치한 python(~\AppData\Local\Programs\Python\Python39\Python.exe)도 보입니다.

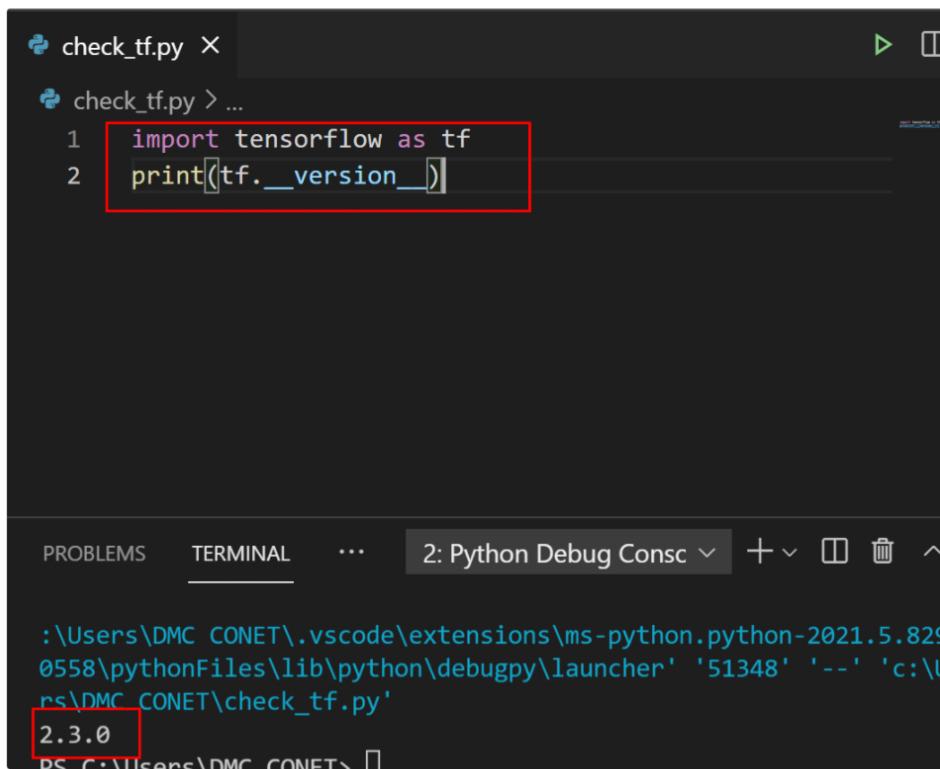
이중에 my\_env의 것을 클릭.



이제 VSCode는 가상환경 my\_env를 사용합니다.

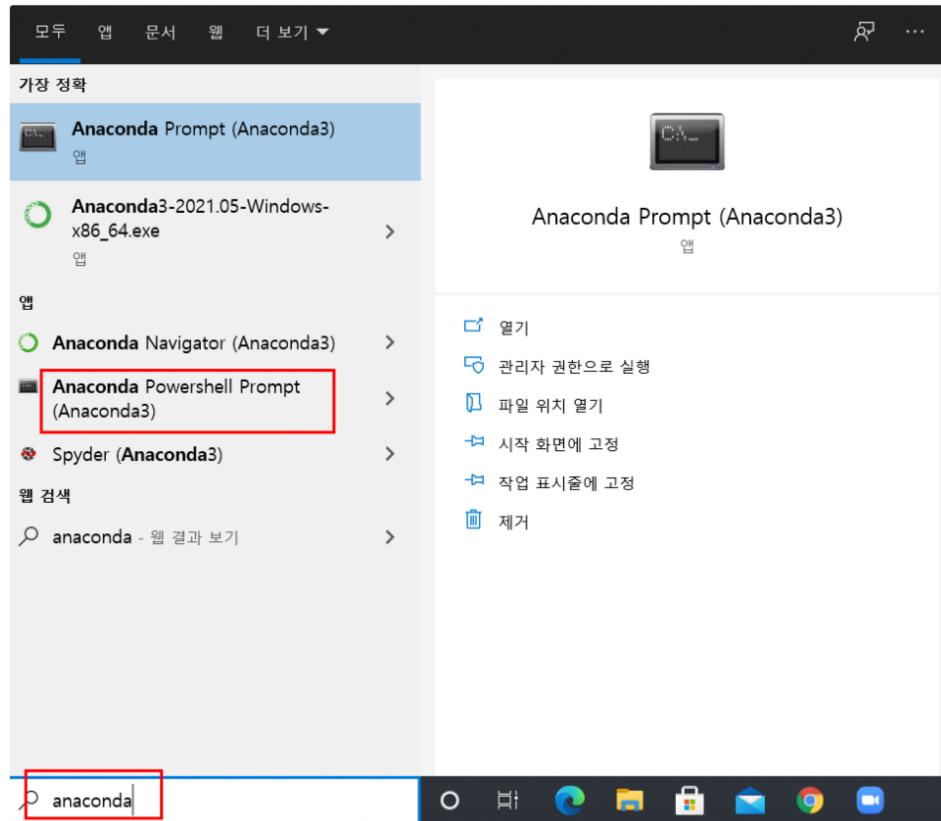
my\_env에 설치된 라이브러리들을 사용할 수 있습니다.

다음과 같이 tensorflow 임포팅 문을 작성하고 실행하면 my\_env에 설치한 tensorflow가 잘 임포팅 됨을 확인 할 수 있습니다.



## 가상 환경과 Jupyter 연동

'Anaconda Powershell Prompt (Anaconda3)'을 실행.



쉘에서 대상 환경으로 들어감.

```
1 conda activate my_env
```

다음 명령으로 ipykernel 설치

```
1 pip insall ipykernel
```

```
(my_env) PS C:\Users\HDCMC CONETS> pip install ipykernel
Collecting ipykernel
  Using cached ipykernel-5.5.5-py3-none-any.whl (120 kB)
Collecting ipython>=5.0.0
  Using cached ipython-7.23.1-py3-none-any.whl (785 kB)
Collecting jupyter-client
  Using cached jupyter_client-6.1.12-py3-none-any.whl (112 kB)
Collecting tornado>=4.2
  Downloading tornado-6.1-cp38-cp38-win_amd64.whl (422 kB)
Collecting traitlets>=4.1.0
  Using cached traitlets-5.0.5-py3-none-any.whl (100 kB)
Collecting jedi>=0.16
  Using cached jedi-0.18.0-py2.py3-none-any.whl (1.4 MB)
Collecting pickleshare
  Using cached pickleshare-0.7.5-py2.py3-none-any.whl (6.9 kB)
```

쥬피터가 사용할 커널에 환경 추가

```
1 python -m ipykernel install --user --name my_env --display-name my_env
```

```
(my_env) PS C:\Users\DMC CONET> python -m ipykernel install --user --name my_env --display-name my_env
Installed kernelspec my_env in C:\Users\DMC CONET\AppData\Roaming\jupyter\kernels\my_env
(my_env) PS C:\Users\DMC CONET>
```

그리고 쥬피터 실행

```
1 jupyter-notebook
```

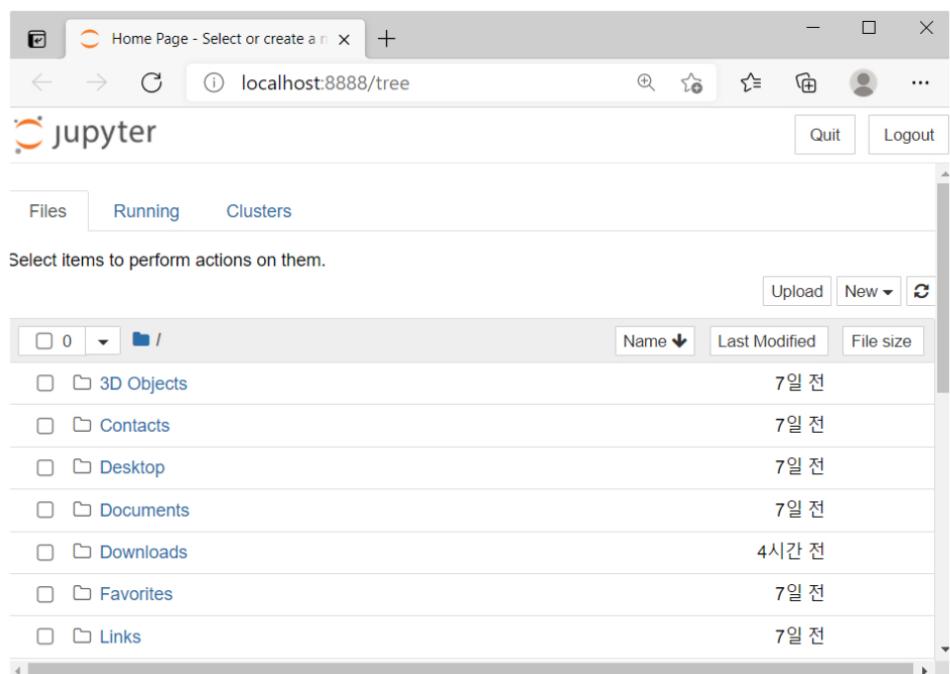
  

```
■ Anaconda Powershell Prompt (Anaconda3)
(base) PS C:\Users\DMC CONET> conda env list
# conda environments:
#
base          * C:\Anaconda3
my_env        C:\Anaconda3\envs\my_env

(base) PS C:\Users\DMC CONET> conda activate my_env
(my_env) PS C:\Users\DMC CONET> jupyter-notebook
[1] 2021-05-18 15:53:29.649 LabApp JupyterLab extension loaded from C:\Anaconda3\lib\site-packages\jupyterlab\extension.py
[1] 2021-05-18 15:53:29.649 LabApp JupyterLab application directory is C:\Anaconda3\share\jupyter\lab
[1] 15:53:29.654 NotebookApp] Serving notebooks from local directory: C:\Users\DMC CONET
[1] 15:53:29.654 NotebookApp] Jupyter Notebook 6.3.0 is running at:
[1] 15:53:29.654 NotebookApp] http://localhost:8888/?token=455c6dff8537dbc14a9aa402f7150024f94
[1] 15:53:29.654 NotebookApp] or http://127.0.0.1:8888/?token=455c6dff8537dbc14a9aa402f7150024f94
[1] 15:53:29.654 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice)
[1] 15:53:29.700 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/DMC%20CONET/AppData/Roaming/jupyter/runtime/nbserver-13260-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=455c6dff8537dbc14a9aa402f7150024f9420265e83c801e
or http://127.0.0.1:8888/?token=455c6dff8537dbc14a9aa402f7150024f9420265e83c801e
```

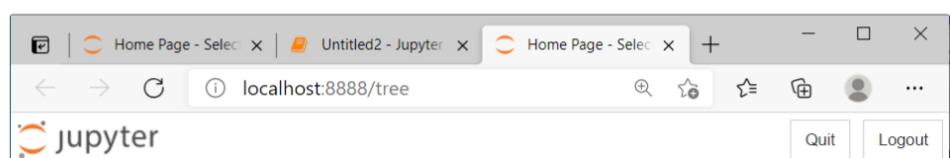
웹브라우저 창이 뜬다.



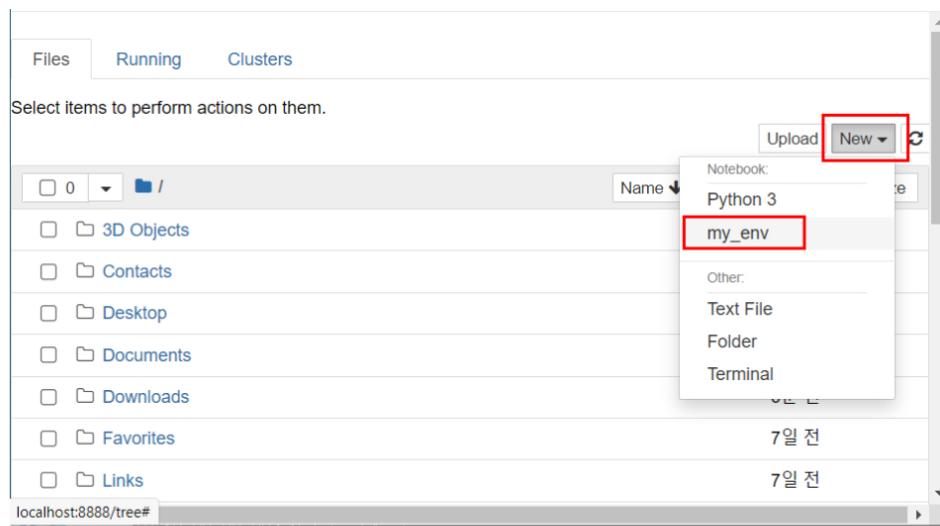
The screenshot shows a web browser window with the title "localhost:8888/tree". The page displays the Jupyter Notebook interface. At the top, there are navigation buttons for back, forward, and search. Below the header, there are two tabs: "Files" (selected) and "Running". A message says "Select items to perform actions on them." Below this, there is a table listing files and folders:

	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	3D Objects	7일 전	
<input type="checkbox"/>	Contacts	7일 전	
<input type="checkbox"/>	Desktop	7일 전	
<input type="checkbox"/>	Documents	7일 전	
<input type="checkbox"/>	Downloads	4시간 전	
<input type="checkbox"/>	Favorites	7일 전	
<input type="checkbox"/>	Links	7일 전	

'New' > 'my\_env' 선택으로 새 노트북 생성.



The screenshot shows a web browser window with three tabs: "Home Page - Select", "Untitled2 - Jupyter", and "Home Page - Select". The active tab is "Untitled2 - Jupyter", which displays the Jupyter Notebook interface at "localhost:8888/tree". The page structure is identical to the previous screenshot, showing the sidebar with file and folder listings.



새 노트북을 만들고 다음 코드를 입력 실행하여 tensorflow가 임포팅 되는지 확인.

localhost:8888/notebooks/Untitled3.ipynb

jupyter Untitled3

File Edit View Insert Cell Kernel Widgets Help Trusted my\_env

In [2]:

```
import tensorflow as tf
print(tf.__version__)
```

2.3.0

In [ ]:

좋아요 처음으로 좋아하는 사람이 되어 볼까요?

레이블 없음



댓글 쓰기...