

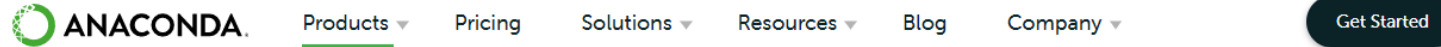


# 머신러닝, 딥러닝 Pytorch & Keras Installation Guide

# Anaconda Virtual Environment

PIAI Research Department

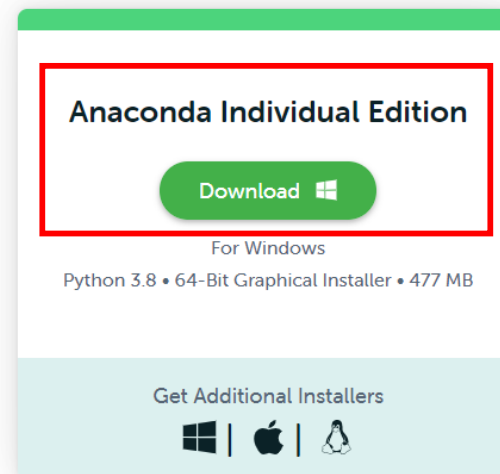
1. <https://www.anaconda.com/products/individual> 접속
2. Anaconda Individual Edition 다운로드



Individual Edition

## Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

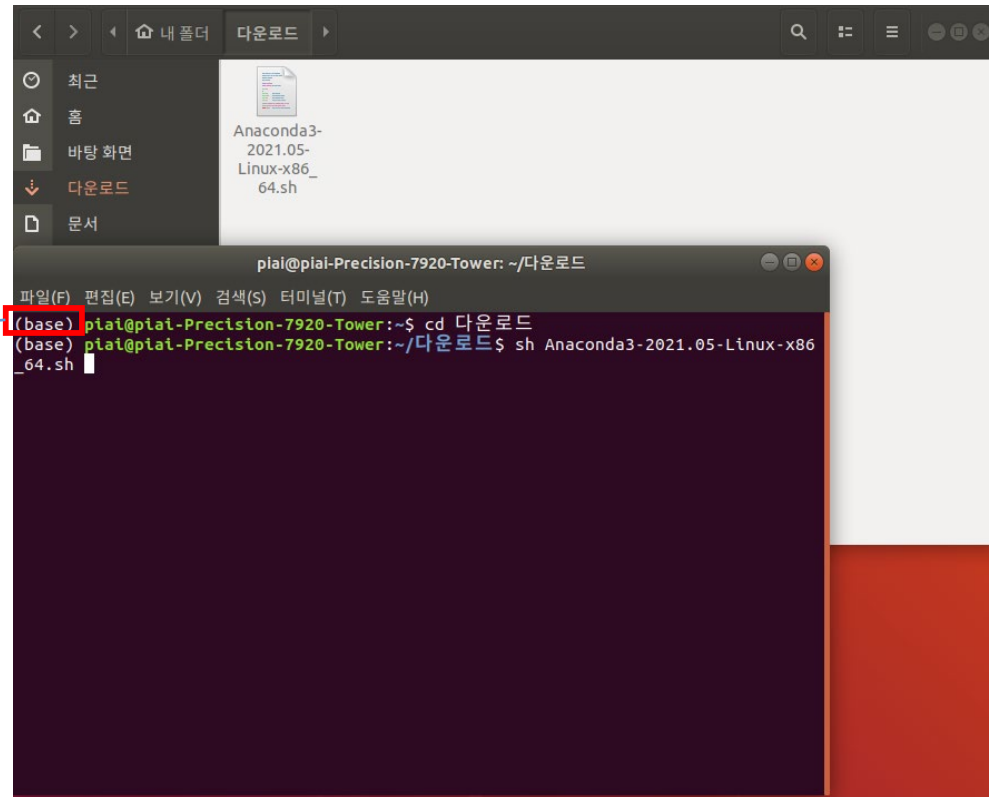


# Anaconda Virtual Environment

PIAI Research Department

1. 터미널에서 Anaconda 설치파일이 다운로드 된 디렉토리로 이동 cf.) cd command
2. sh 명령어로 Anaconda 설치파일 실행

설치 완료 시 터미널 좌측  
에 (base) 표시가 나타남



The screenshot shows a file manager window with a sidebar on the left containing icons for '최근' (Recent), '홈' (Home), '바탕 화면' (Desktop), '다운로드' (Downloads), and '문서' (Documents). The main area displays a file named 'Anaconda3-2021.05-Linux-x86\_64.sh'. Overlaid on this is a terminal window titled 'piaI@piaI-Precision-7920-Tower: ~/다운로드'. The terminal shows the following commands and output:

```
(base) piaI@piaI-Precision-7920-Tower:~$ cd 다운로드
(base) piaI@piaI-Precision-7920-Tower:~/다운로드$ sh Anaconda3-2021.05-Linux-x86_64.sh
```

A red box highlights the '(base)' prompt in the first line of the terminal output, and a blue arrow points from the Korean text box to this prompt.

# Anaconda Virtual Environment

PIAI Research Department

## 1. 아나콘다 설치 후 업데이트(필수!)

- conda update -n base conda
- conda update --all
- python -m pip install --upgrade pip
- conda --version

(2023.03.30 기준 23.3.0)

# Anaconda Virtual Environment

PIAI Research Department

## 1. 가상환경 생성 - 사용할 파이썬 버전 = 3.8

- conda create -n <your environment> **python = 3.8** → (base) pirl@pirl-PowerEdge-R740:~\$ conda create -n mldl python=3.8

## 2. 가상환경 활성화 - 실습 시 반드시 생성한 가상환경을 활성화 한 상태로 진행할 것

- conda activate <your environment> → (base) pirl@pirl-PowerEdge-R740:~\$ conda activate mldl  
(mldl) pirl@pirl-PowerEdge-R740:~\$

## 3. 파이썬 버전 확인

- python --version → (mldl) pirl@pirl-PowerEdge-R740:~\$ python --version  
Python 3.8.13

## 4. 가상환경 비활성화 - 가상환경 비활성화 할 때만 사용할 것

- conda deactivate → (mldl) pirl@pirl-PowerEdge-R740:~\$ conda deactivate  
(base) pirl@pirl-PowerEdge-R740:~\$

# Anaconda Virtual Environment

PIAI Research Department

## 5. 주피터 노트북 설치 - 새로운 가상환경에 주피터 노트북을 설치해 줘야 함

- conda install jupyter notebook

```
(mldl) pirl@pirl-PowerEdge-R740:~$ conda install jupyter notebook
```

## 6. 새로 생성한 가상환경을 주피터 노트북 커널로 등록

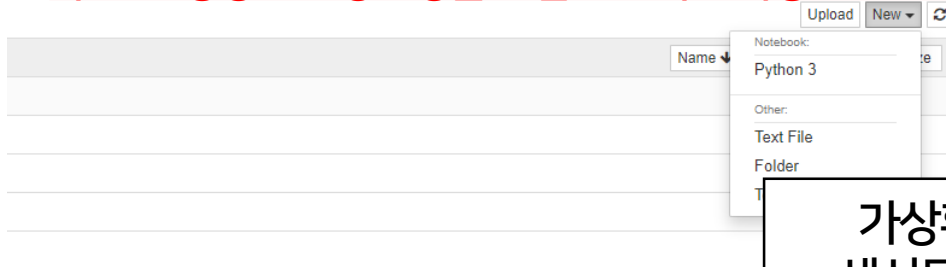
- pip install ipykernel

- python -m ipykernel install --user --name 가상환경이름

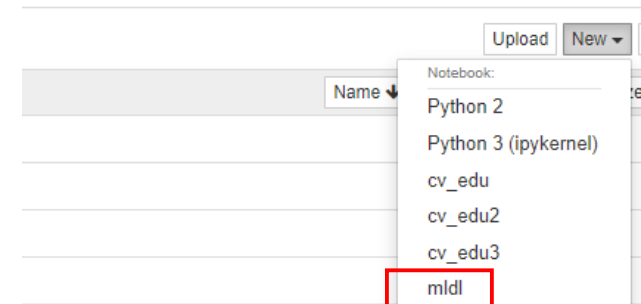
```
(mldl) pirl@pirl-PowerEdge-R740:~$ python -m ipykernel install --user --name mldl
Installed kernelspec mldl in /home/pirl/.local/share/jupyter/kernels/mldl
```

※ 새 .ipynb 파일 생성 시 커널 설정 방법

→ 꼭 새로 생성한 가상환경을 커널로 선택해 작동시켜야 함



가상환경 등록 시 mldl이  
생성된 것을 확인할 수 있음



# Anaconda Virtual Environment

PIAI Research Department

※ .ipynb 파일 안에서 커널 변경하는 방법

→ 꼭 새로 생성한 가상환경을 커널로 선택해 작동시켜야 함

The screenshot shows the JupyterLab interface with the 'Kernel' menu open. The 'Change kernel' option is selected, displaying a list of available kernels. The 'mldl' kernel is highlighted in the list. A red box highlights the 'mldl' kernel in the list, and another red box highlights the 'mldl' kernel in the top right corner of the interface. A red arrow points from the highlighted 'mldl' kernel in the list to the highlighted 'mldl' kernel in the top right corner. A text box with the Korean text '클릭시 커널 변경됨' (Click to change kernel) is positioned between the two highlighted 'mldl' kernels.

jupyter Untitled Last Checkpoint: 2시간 전 (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted mldl

Interrupt Restart Restart & Clear Output Restart & Run All Reconnect Shutdown

Change kernel

- Python 2
- Python 3 (ipykernel)
- cv\_edu
- cv\_edu2
- cv\_edu3
- mldl**

클릭시 커널 변경됨

# Check/Install NVIDIA GPU DRIVER

PIAI Research Department

## 1. 현재 GPU Driver 확인

- nvidia-smi

```
(cv_edu) pirl@pirl-PowerEdge-R740:~$ nvidia-smi
Fri Jun 17 16:58:53 2022
```

NVIDIA-SMI 470.57.02				Driver Version: 470.57.02				CUDA Version: 11.4			
GPU Fan	Name Temp	Persistence-M Perf	Bus-Id Pwr:Usage/Cap	Disp.A Memory-Usage	Volatile GPU-Util	Uncorr. Compute	ECC M.				
0	Tesla V100-PCIE...	On	00000000:3B:00.0	Off	0						
N/A	28C	P0	24W / 250W	0MiB / 16160MiB	0%	Default	N/A				

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory	Usage
ID	ID	ID				Usage	
No running processes found							

1. nvidia-smi 명령어 실행 시 그림과 같이 출력되나 **드라이버 버전이 470 이상이 아닌 경우**
2. nvidia-smi 명령어 실행 시 그림과 같이 출력되지 않고 **해당 명령어가 없다고 뜨는 경우**

- sudo apt-get install nvidia-driver-470

컴퓨터 리부트!

※ 드라이버 설치 후 재부팅을 하지 않으면 환경에 문제가 생깁니다.

※ 드라이버 버전은 반드시 470 이상이어야 합니다.



# Install CUDA/cuDNN/Tensorflow/Pytorch

PIAI Research Department

## 1. CUDA / CuDNN 설치

- conda install -c conda-forge cudatoolkit=11.3 cudnn=8.1

## 2. tensorflow-gpu 설치 - 2.6.0 V / 반드시 GPU 버전으로 설치할 것

- pip install tensorflow-gpu==2.6.0

GPU

버전	Python 버전	컴파일러	빌드 도구	cuDNN	CUDA
tensorflow-2.7.0	3.7~3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.6.0	3.6~3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.5.0	3.6~3.9	GCC 7.3.1	Bazel 3.7.2	8.1	11.2
tensorflow-2.4.0	3.6-3.8	GCC 7.3.1	Bazel 3.1.0	8.0	11.0
tensorflow-2.3.0	3.5-3.8	GCC 7.3.1	Bazel 3.1.0	7.6	10.1
tensorflow-2.2.0	3.5-3.8	GCC 7.3.1	Bazel 2.0.0	7.6	10.1
tensorflow-2.1.0	2.7, 3.5-3.7	GCC 7.3.1	Bazel 0.27.1	7.6	10.1
tensorflow-2.0.0	2.7, 3.3-3.7	GCC 7.3.1	Bazel 0.26.1	7.4	10.0

# Install CUDA/cuDNN/Tensorflow/Pytorch

PIAI Research Department

tensorflow에서 GPU 인식이 되는지 확인  
터미널에서 파이썬 실행 후 또는 주피터노트북에서 아래의 코드를 실행할 것

- from tensorflow.python.client import device\_lib
- device\_lib.list\_local\_devices()

→ physical\_device\_desc에 내 PC의 GPU명이 출력되면 인식  
이 되는 것

- A,B반 : RTX 2080 SUPER GPU 인식
- C반 : RTX A6000 GPU 인식

## - 명령어 실행

```
(cv_edu) pirl@pirl-PowerEdge-R740:~$ python
Python 3.8.13 (default, Mar 28 2022, 11:38:47)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from tensorflow.python.client import device_lib
2022-06-17 17:06:01.751902: I tensorflow/stream_executor/platform/default
ader.cc:49] Successfully opened dynamic library libcudart.so.11.0
>>> device_lib.list_local_devices()
```

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 6427799706701115374
], name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 15478334464
locality {
  bus_id: 1
  links {
  }
}
incarnation: 4777911059069591752
physical_device_desc: "device: 0, name: Tesla V100-PCIE-16GB, pci bus id: 0000:3
b:00.0, compute capability: 7.0"
]
```

이것만 표출될 시 GPU 인식이 안되는 것  
이므로 재설치할 것

- 만약 GPU 명이 출력되지 않을 시 가상환경을 삭제하고 처음 부터 만든 후 다시 설치를 진행

※ 가상환경을 지우는 명령어 :

- jupyter kernelspec uninstall 가상환경이름
- conda remove -n 가상환경이름 -all
- 이후 처음부터 가상환경을 만들어 진행할 것

# Install CUDA/cuDNN/Tensorflow/Pytorch

PIAI Research Department

## 1. Pytorch 설치 - 1.10.1V

- conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 -c pytorch

```
(mldl) pirl@pirl-PowerEdge-R740:~$ conda install pytorch==1.10.1 torchvision==0.11.2 torchaudio==0.10.1 -c pytorch
```

## 2. GPU Available Check & GPU Device Name 확인

python 실행 후 또는 주피터 노트북에서 아래 코드 실행

- import torch
- torch.cuda.is\_available() → True 출력 시 정상 설치 완료
- torch.cuda.get\_device\_name(0) → GPU 명 출력 시 확인

- A,B반 : RTX 2080 SUPER GPU 인식
- C반 : RTX A6000 GPU 인식

```
(mldl) pirl@pirl-PowerEdge-R740:~$ python
Python 3.8.13 (default, Mar 28 2022, 11:38:47)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.cuda.get_device_name(0)
'Tesla V100-PCIE-16GB'
```

- 만약 False 또는 GPU 명이 출력되지 않을 시 가상환경을 삭제하고 처음 부터 만든 후 다시 설치를 진행

# Install Required Packages

PIAI Research Department

- 미리 설치해 두어야 할 패키지 목록
  - `conda install -c conda-forge graphviz`
  - `conda install matplotlib scipy tqdm`
  - `pip install pydot seaborn sklearn scikit-image torchsummary opencv-python split-folders`
  - `conda install scikit-learn`

# Start Jupyter Notebook

PIAI Research Department

- 주피터 노트북을 켤 때 유의사항

1. 반드시 터미널에서 'conda activate 가상환경이름'으로 가상환경을 구동시킨 후에
2. 'jupyter notebook'을 입력하여 주피터 노트북을 켜주도록 합니다.

```
(ml) C:\Users\hnyan>jupyter notebook
```

3. 그렇게 하지 않고 (base) 환경에서 주피터 노트북을 켜게 되면 가상환경이 꼬이면서 코드 실행 시 에러가 발생합니다.



# 감사합니다.

설치 오류 및 문의 사항

A반 : 여동훈 연구원(tgc01098@postech.ac.kr)

B반 : 권현섭 연구원(kert06@postech.ac.kr)

C반 : 이창엽 연구원(lcy8417@postech.ac.kr)