

세션 2

실습을 위한 환경설정 안내

1. 아나콘다 설치

2. 가상환경 생성 / 실행

3. Pytorch 설치

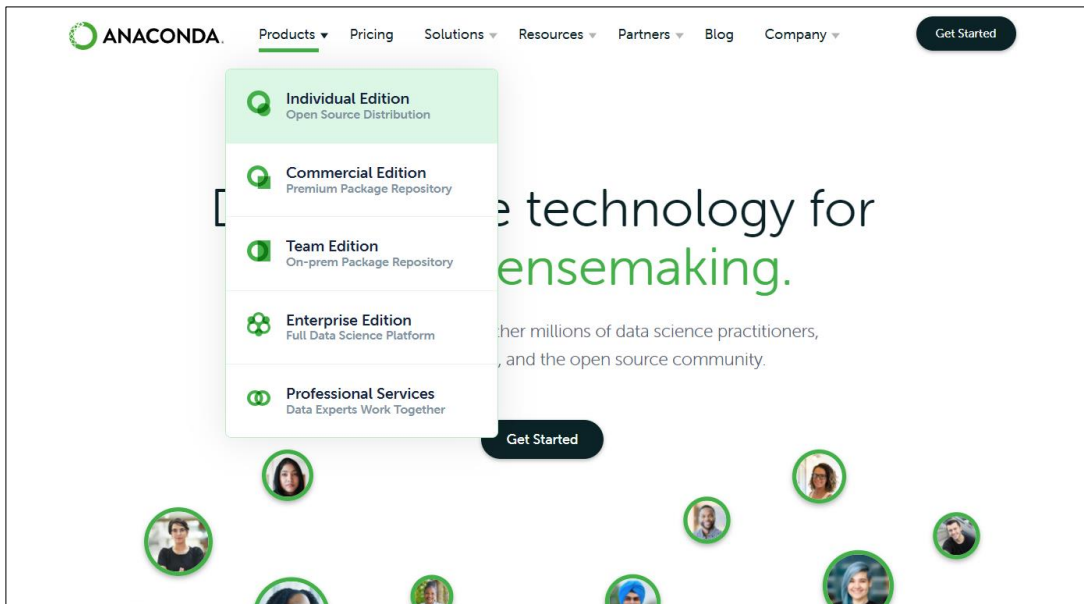
4. 기타 패키지 설치

5. Jupyter notebook 실행

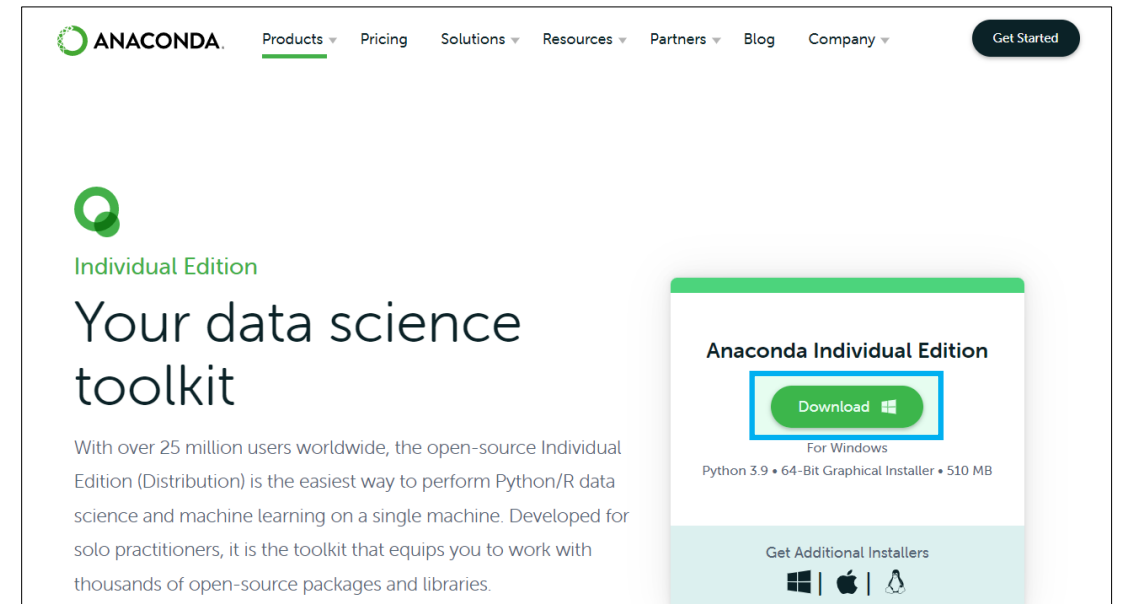
1.1 아나콘다 설치 프로그램 다운받기

1.1.1 웹브라우저에서 아나콘다 홈페이지 접속 (anaconda.com)

1.1.2 'Products > Individual Edition' 메뉴 선택



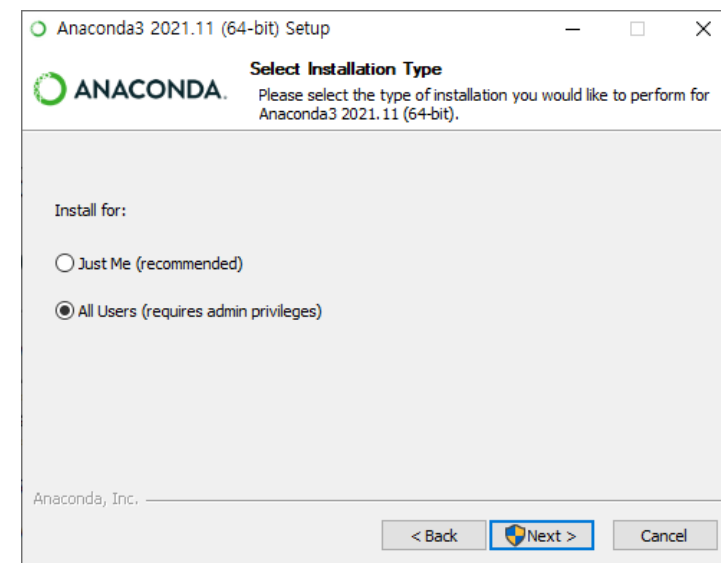
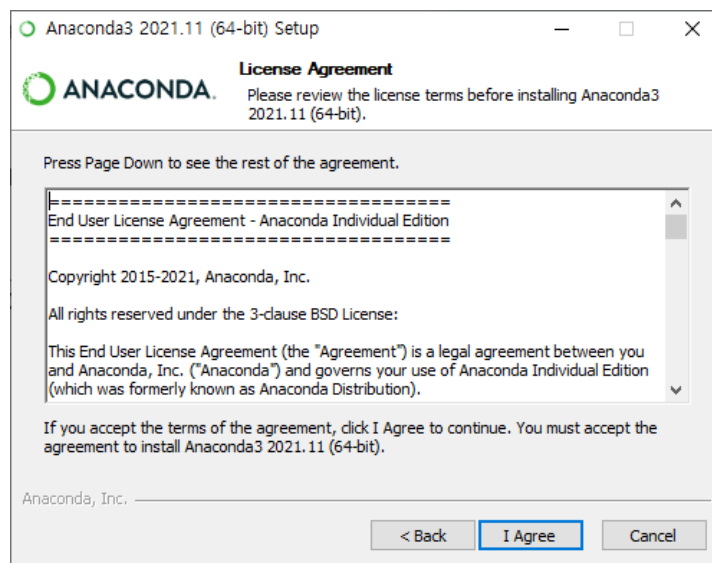
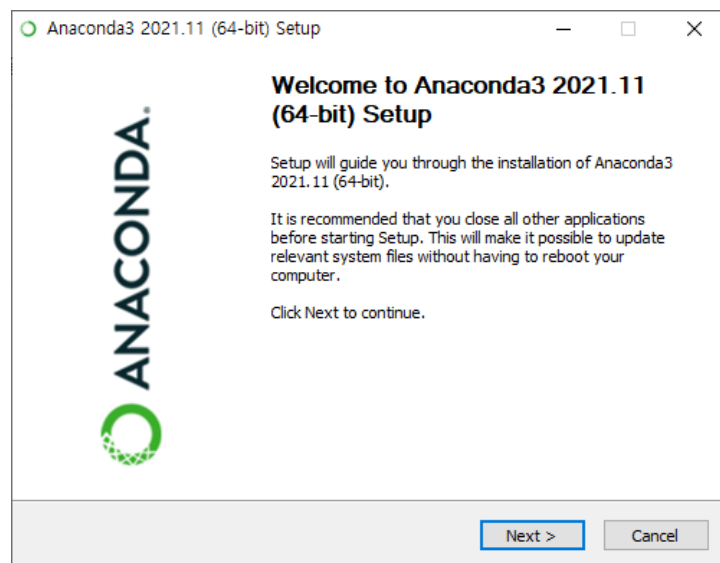
1.1.3 'Download' 버튼 클릭



1.2 Anaconda 설치

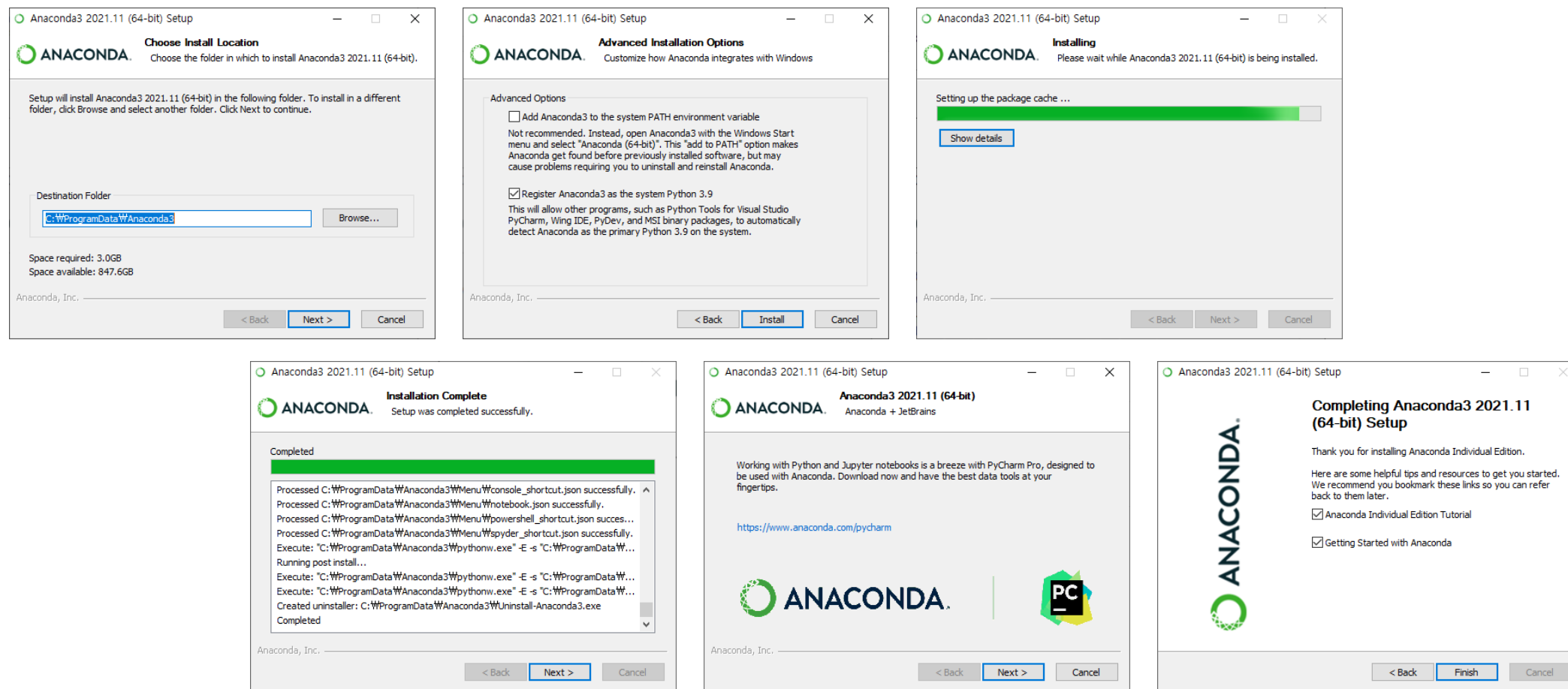
1.2.1 'Anaconda3-2021.11-Windows-x86_64.exe' 클릭

1.2.2 Anaconda 설치 프로그램의 설치 안내에 따라 설치 (1)



1.2 Anaconda 설치

1.2.2 Anaconda 설치 프로그램의 설치 안내에 따라 설치 (2)

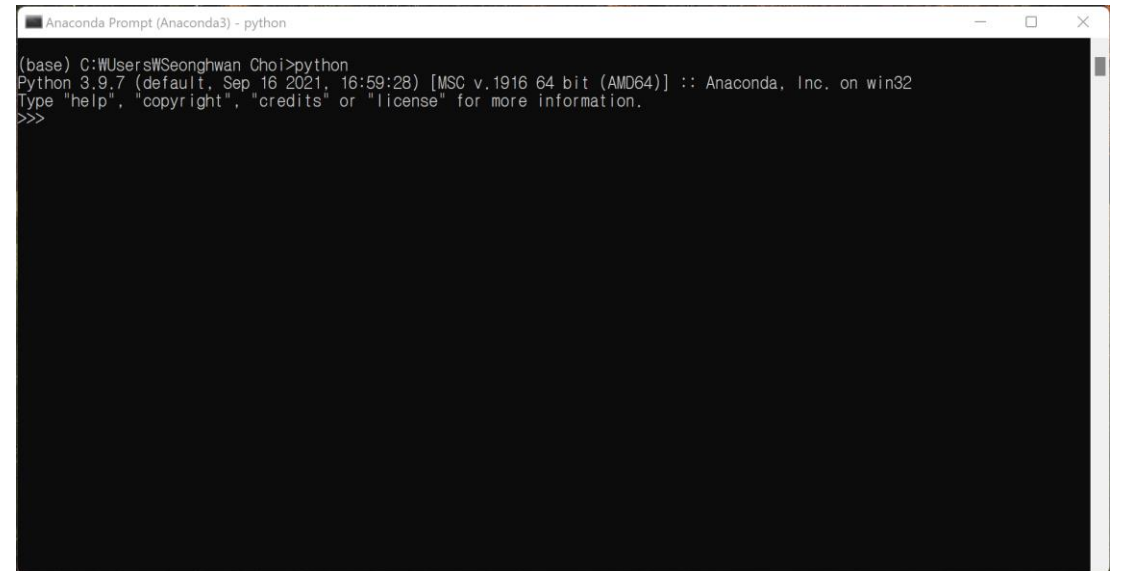
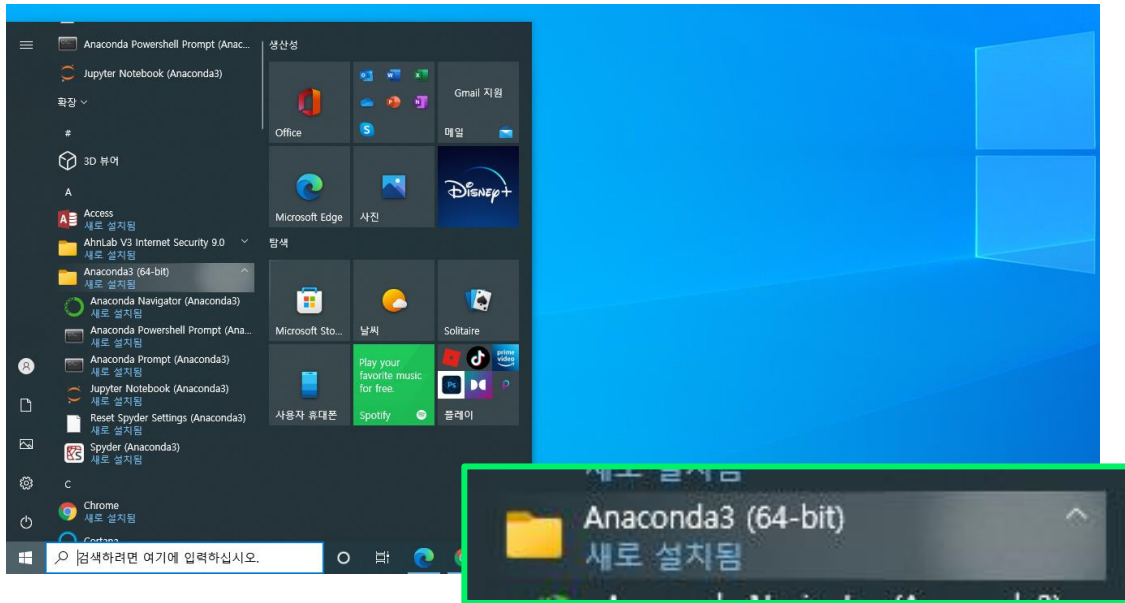


1.3 Anaconda 설치 확인하기

1.3.1 윈도우 시작(Start) 메뉴에서 'Anaconda 3' 설치 확인

1.3.2 Anaconda Prompt (Anaconda 3) 실행

1.3.3 'python' 실행

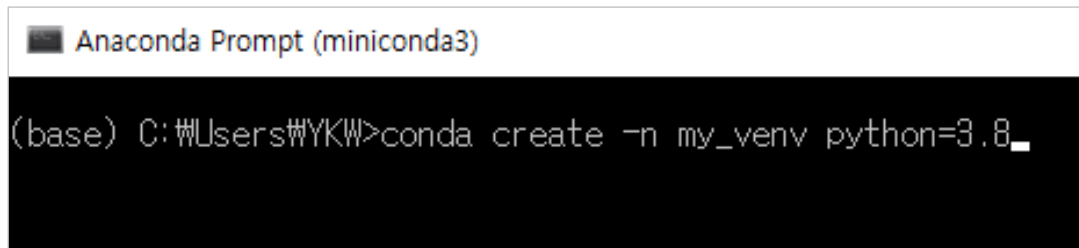


1. **아나콘다 설치**
2. **가상환경 생성 / 실행**
3. **Pytorch 설치**
4. **기타 패키지 설치**
5. **Jupyter notebook 실행**

2.1 가상환경 생성

2.1.1 Anaconda prompt 실행

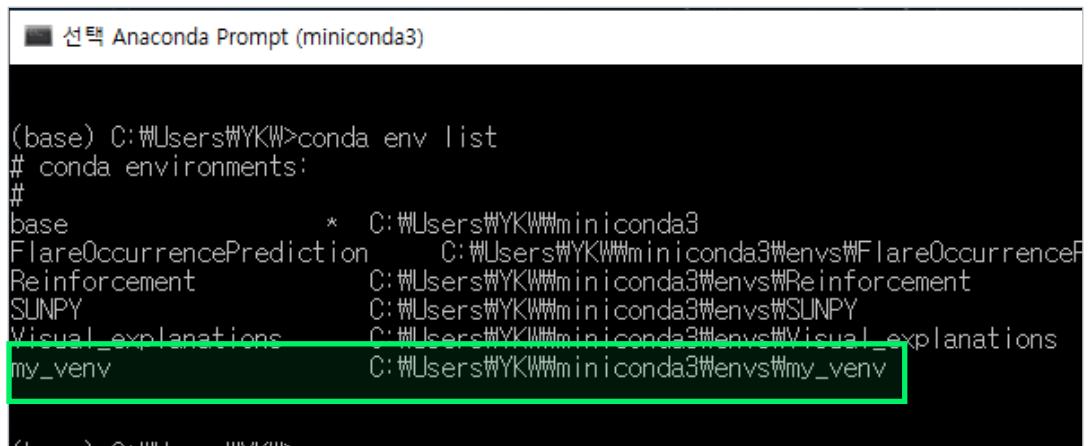
2.1.2 conda create -n [가상환경이름] python=3.8 실행



```
Anaconda Prompt (miniconda3)

(base) C:\Users\YKW>conda create -n my_venv python=3.8_
```

2.1.3 설치 후 conda env list 실행하여 해당 가상환경이 생성되었나 확인

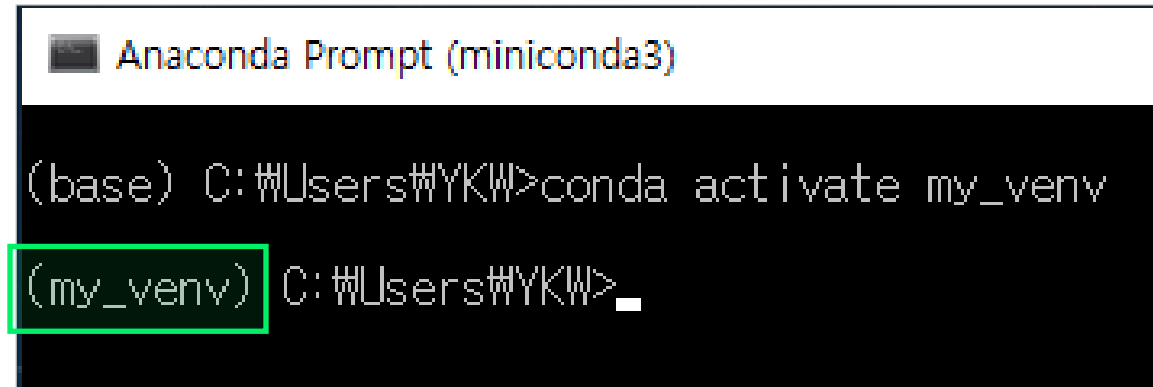


```
선택 Anaconda Prompt (miniconda3)

(base) C:\Users\YKW>conda env list
# conda environments:
#
base                  *  C:\Users\YKW\miniconda3
FlareOccurrencePrediction  C:\Users\YKW\miniconda3\envs\FlareOccurrencePrediction
Reinforcement          C:\Users\YKW\miniconda3\envs\Reinforcement
SUNPY                  C:\Users\YKW\miniconda3\envs\SUNPY
Visual_explanations    C:\Users\YKW\miniconda3\envs\Visual_explanations
my_venv                C:\Users\YKW\miniconda3\envs\my_venv
```

2.2 가상환경 실행

2.2.1 conda activate [가상환경이름] 실행 (입력창 앞부분이 가상환경 이름으로 바뀐 것 확인)



```
Anaconda Prompt (miniconda3)

(base) C:\Users\YK\>conda activate my_venv

(my_venv) C:\Users\YK\>_
```

1. 아나콘다 설치
2. 가상환경 생성 / 실행
3. **Pytorch 설치**
4. 기타 패키지 설치
5. Jupyter notebook 실행

3. Pytorch 설치

3.1 <https://pytorch.org/get-started/locally/> 접속

3.2 위의 사이트에서,

- Python Build : LTS(1.8.2)
- Your OS : 실습자의 OS에 맞게 선택
- Package : Conda
- Language : Python
- Computer Platform : CPU

선택 후 Run this Command 에 생성되는 명령어를 생성한 가상환경에서 실행

* Mac의 경우 CUDA 사용이 어렵기 때문에 실습은 CPU 버전으로 진행

만약 본인의 실습 환경에 nvidia GPU가 있고 CUDA 환경에서 실습하고 싶다면
Pytorch 시 Computer platform에서 cuda 버전을 선택하면 conda가 자동으로 설치함.
(nvidia drive는 설치가 되어 있어야 함)

PyTorch Build	Stable (1.10.2)	Preview (Nightly)	LTS (1.8.2)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
Compute Platform	CUDA 10.2	CUDA 11.1	ROCm 4.2 (beta)	CPU
Run this Command:	<pre>conda install pytorch torchvision torchaudio cpuonly -c pytorch-lts</pre> Note: Additional support for these binaries may be provided by PyTorch Enterprise Support Program Participants.			

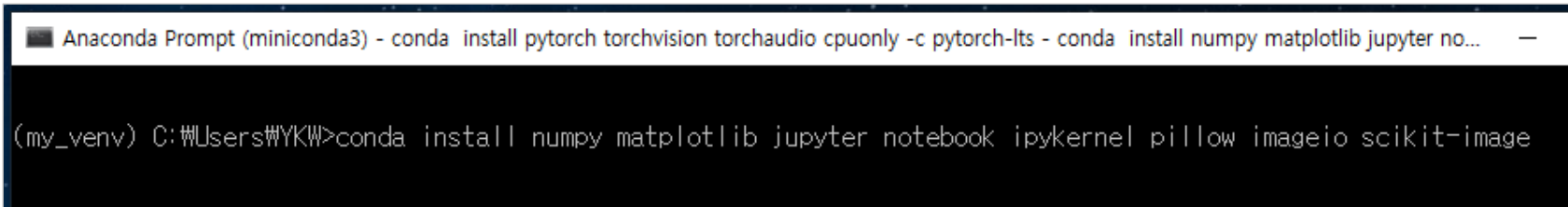
```
Anaconda Prompt (miniconda3)
(my_venv) C:\Users\WYKW>conda install pytorch torchvision torchaudio cpuonly -c pytorch-lts
```

1. 아나콘다 설치
2. 가상환경 생성 / 실행
3. Pytorch 설치
4. **기타 패키지 설치**
5. Jupyter notebook 실행

4. 기타 패키지 설치

필요한 패키지: NumPy, Matplotlib, Jupyter, Notebook, Ipykernel, PIL, imageio, scikit-image

생성된 가상환경에서 `conda install numpy matplotlib jupyter notebook ipykernel pillow imageio scikit-image` 실행



```
Anaconda Prompt (miniconda3) - conda install pytorch torchvision torchaudio cpuonly -c pytorch-lts - conda install numpy matplotlib jupyter no...  
(my_venv) C:\Users\WYK\>conda install numpy matplotlib jupyter notebook ipykernel pillow imageio scikit-image
```

1. 아나콘다 설치
2. 가상환경 생성 / 실행
3. Pytorch 설치
4. 기타 패키지 설치
5. **Jupyter notebook 실행**

5. Jupyter notebook 실행

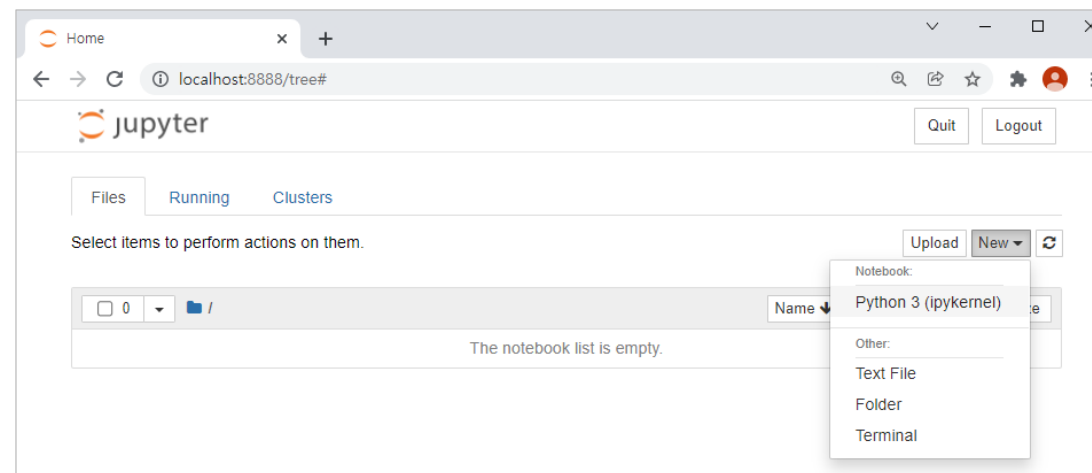
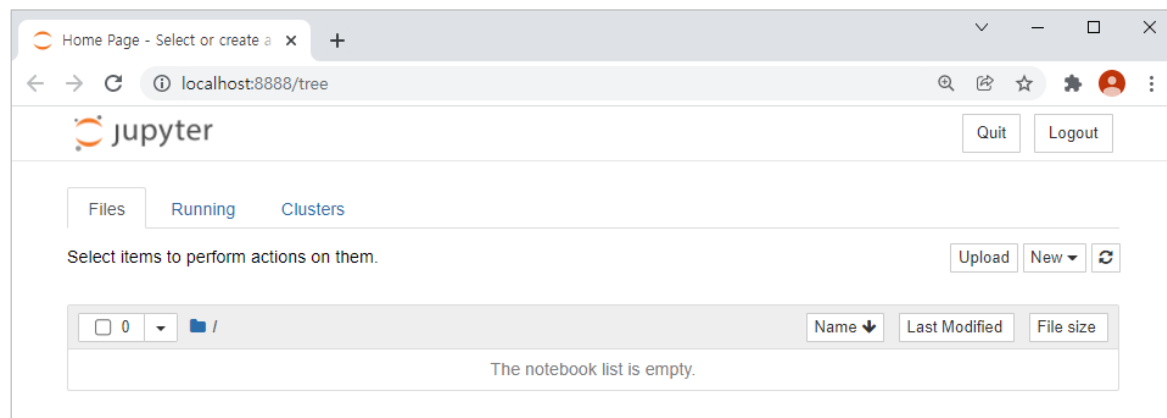
5.1 Anaconda prompt 창에서 cd [파일 경로] 입력하여 강의에 사용하고자 하는 폴더로 이동.

5.2 가상환경에서 Jupyter notebook 실행

```
Anaconda Prompt (miniconda3) - conda install pytorch torchvision torchaudio  
  
(my_venv) C:\Users\YKW>cd c:\Users\YKW\lecture
```

```
Anaconda Prompt (miniconda3) - conda install pytorch torchvision torchaudio  
  
(my_venv) c:\Users\YKW\lecture>jupyter notebook
```

5.3 오른쪽 위 New를 눌러 Python 3 클릭

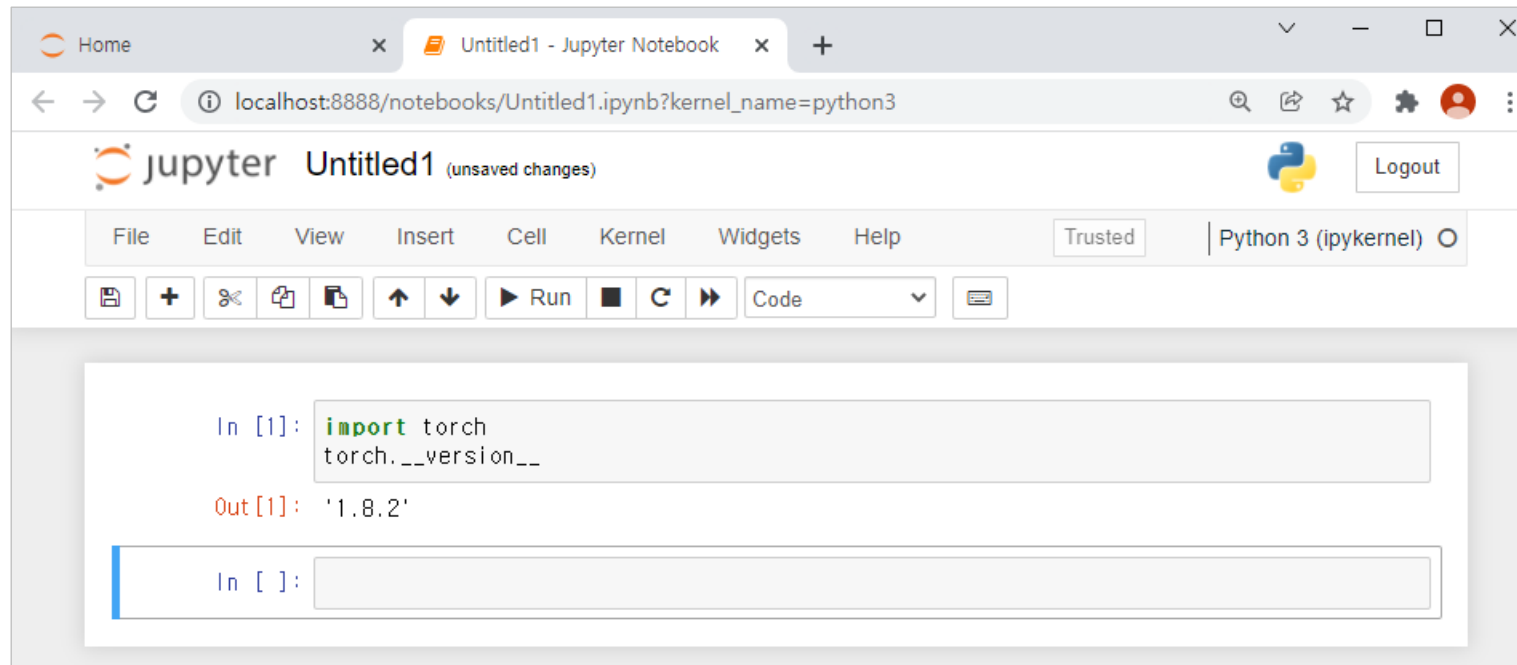


5. Jupyter notebook 실행

5.4 import torch

```
torch.__version__
```

입력 후 shift+Enter 키를 눌러 정상 실행 확인 (설치한 pytorch version 출력 시 이상 없음)



5. Jupyter notebook 실행

5.5 Jupyter Notebook 에서

Import torch

torchvision.models.resnet50(pretrained=True)

사전에 실행하여 모델을 다운받아 오기를 추천 (다른 모델도 가능)

```
In [8]: import torchvision
torchvision.models.resnet50(pretrained=True)

Out [8]: ResNet(
  (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
  (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (layer1): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
  (layer2): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
  (layer3): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
  (layer4): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(1024, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (conv3): Conv2d(128, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
  )
  (fc): Linear(2048, 1000, bias=True)
)
```