

환경 구축

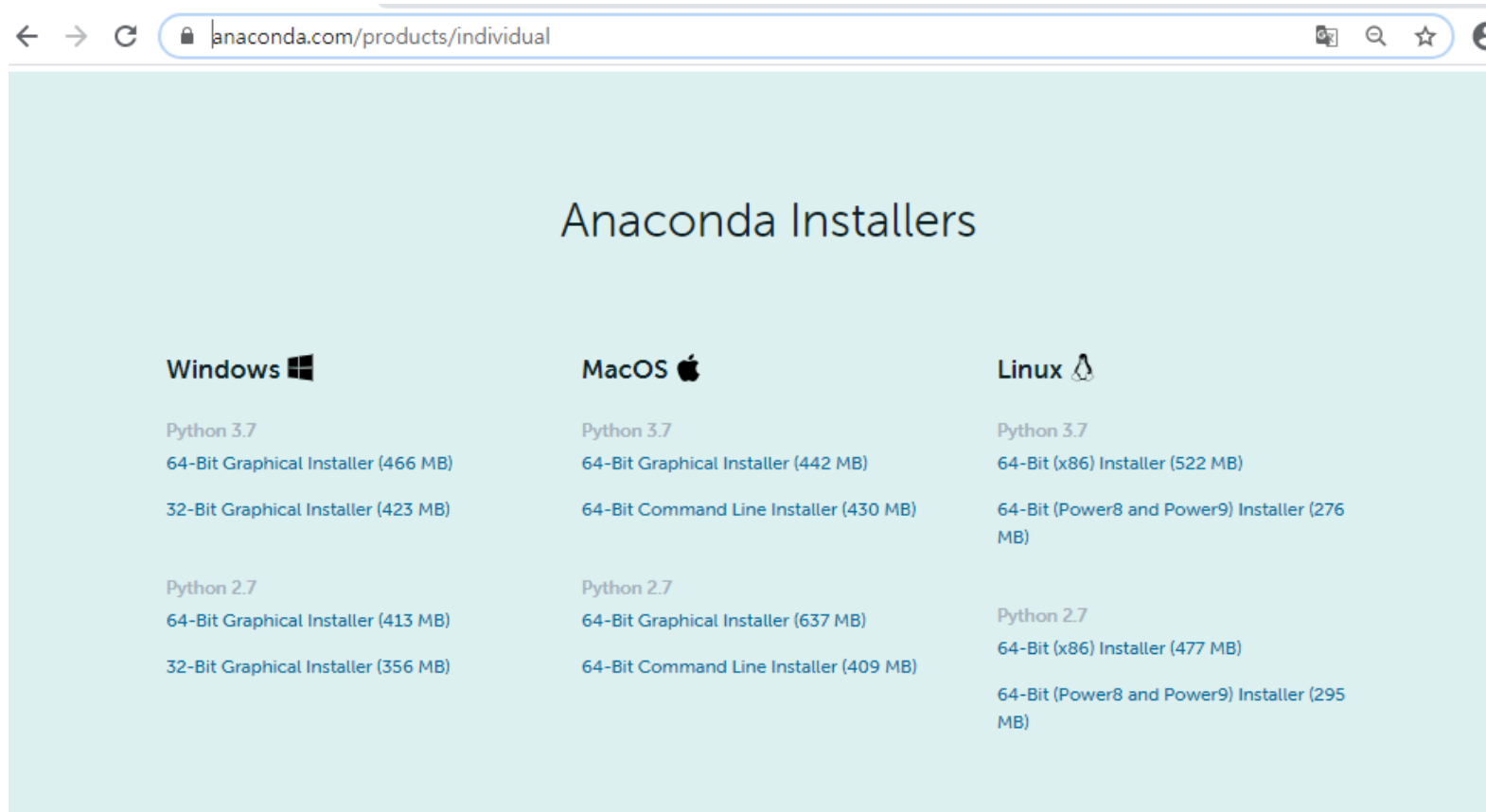


# 아나콘다 설치

아나콘다 URL(<https://www.anaconda.com/products/individual>)

다운로드 : 64-Bit Graphical Installer (466 MB)

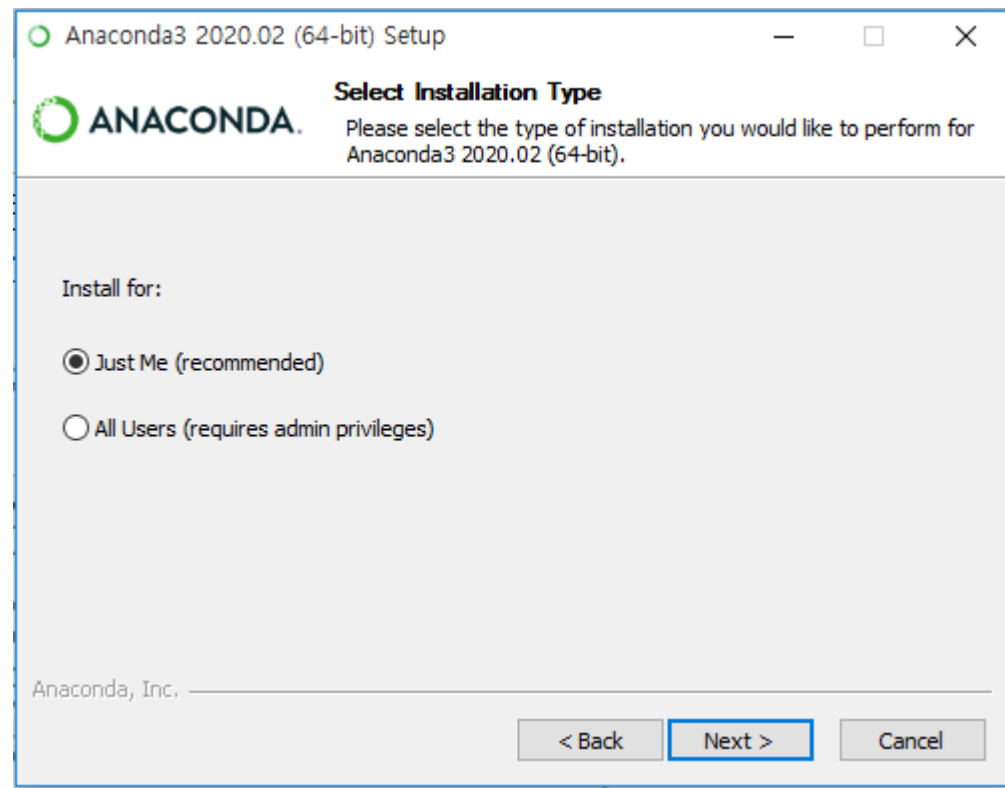
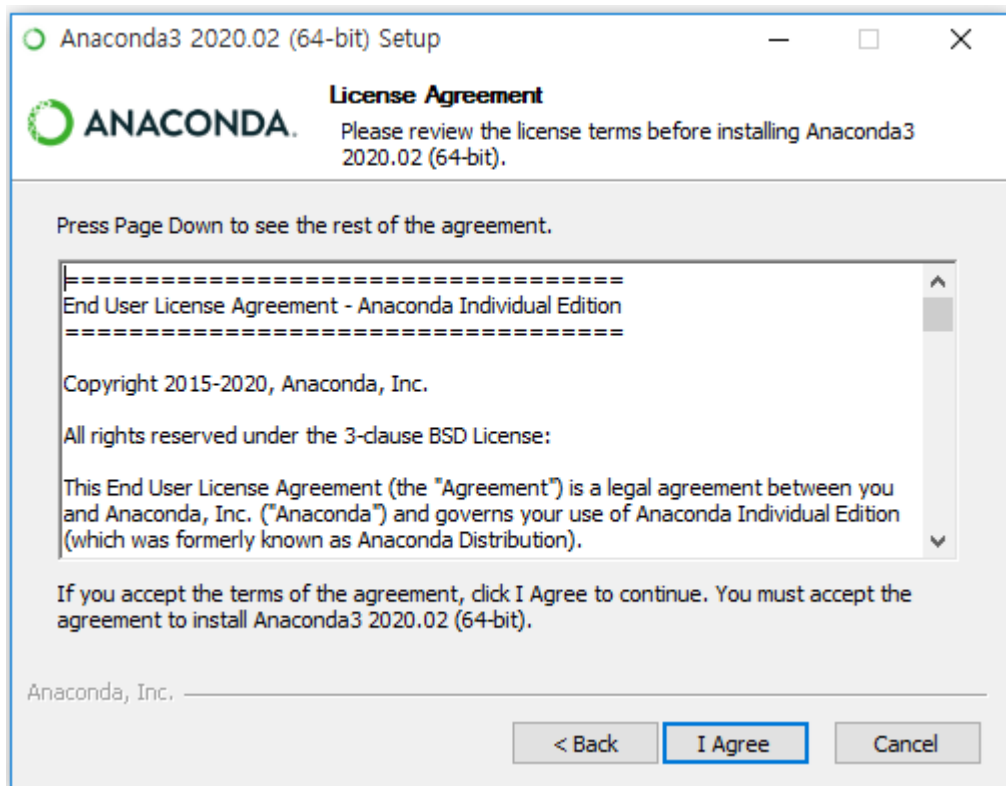
다운로드 후 파일 : Anaconda3-2020.02-Windows-x86\_64



# 아나콘다 설치

아나콘다

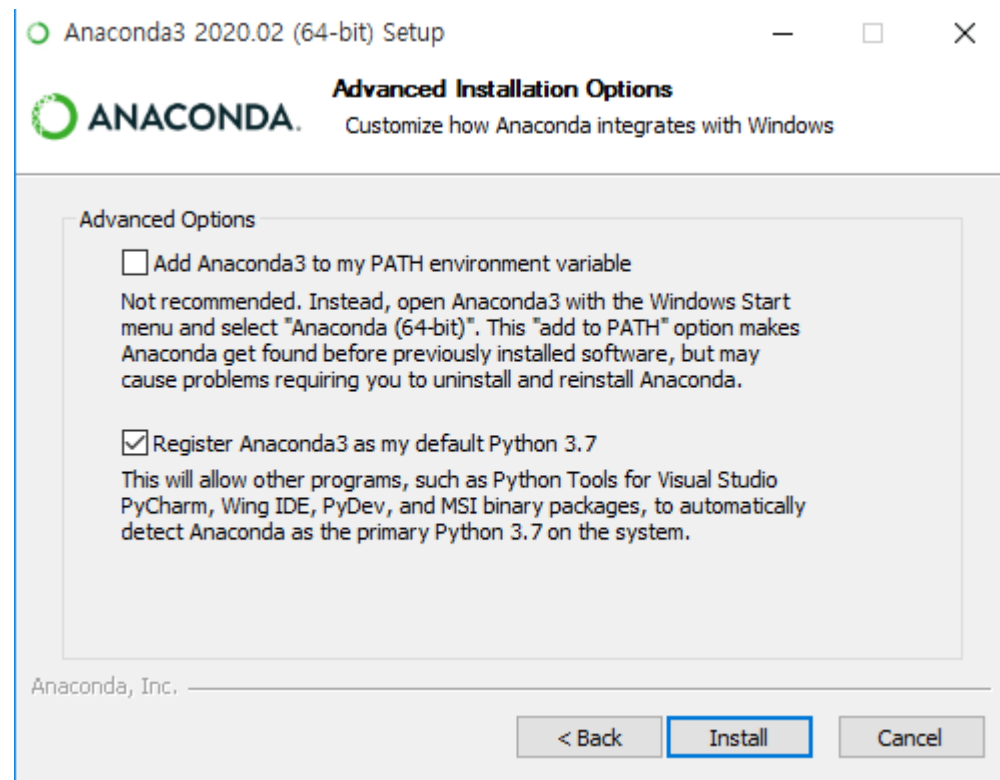
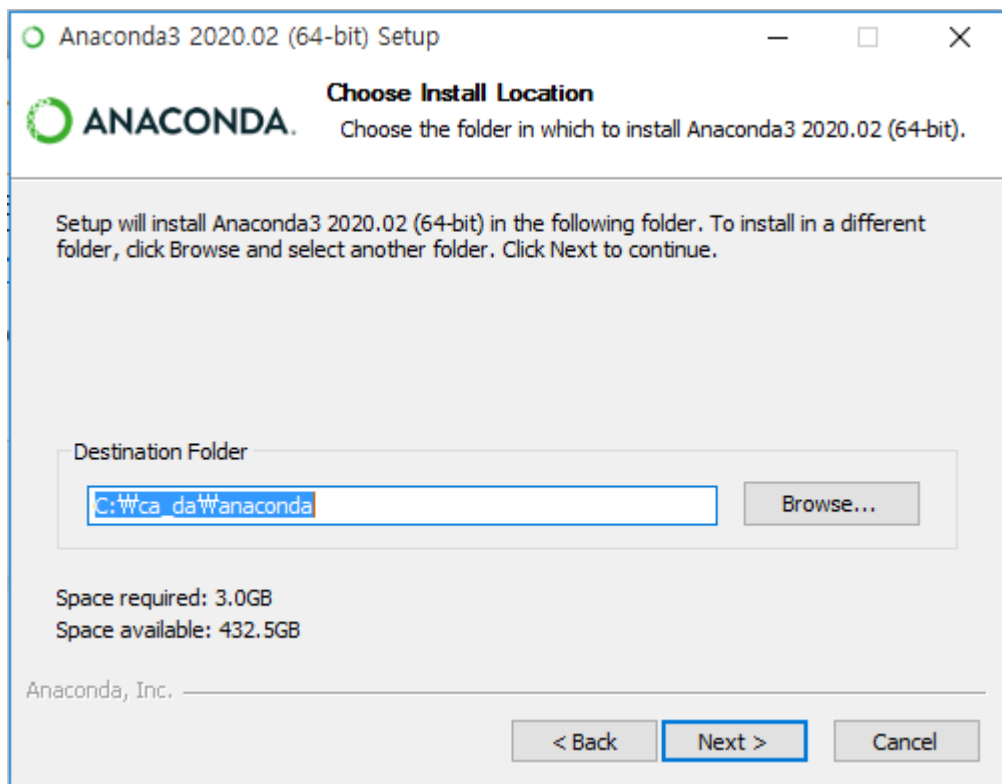
URL(<https://www.anaconda.com/products/individual>)



# 아나콘다 설치

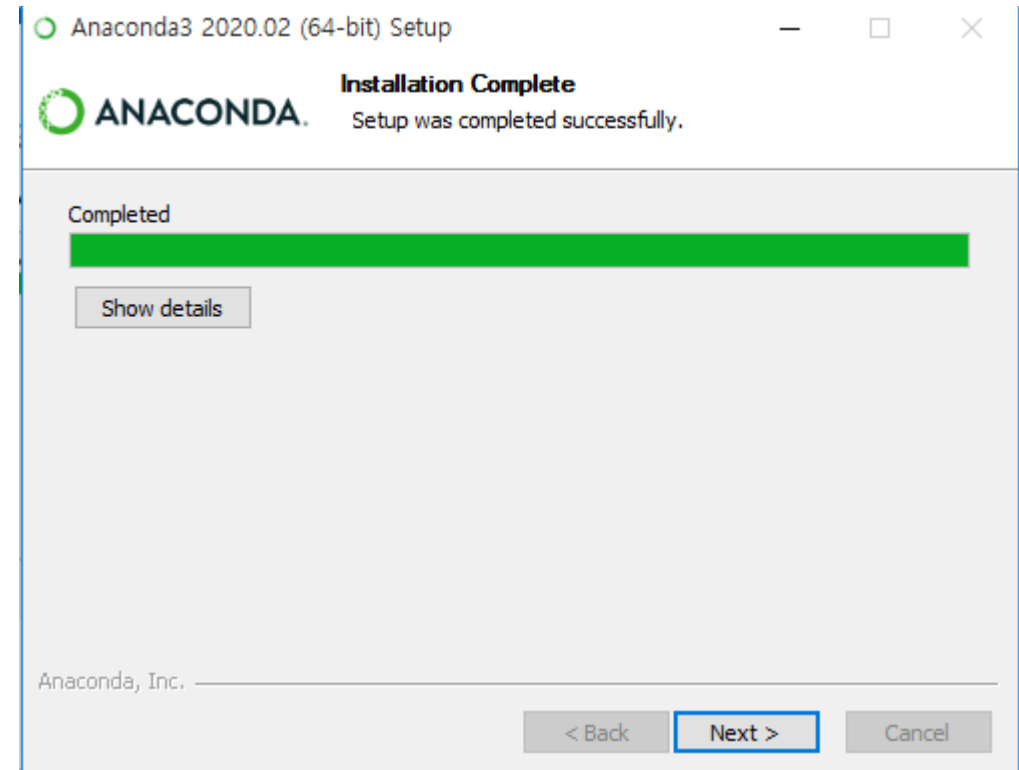
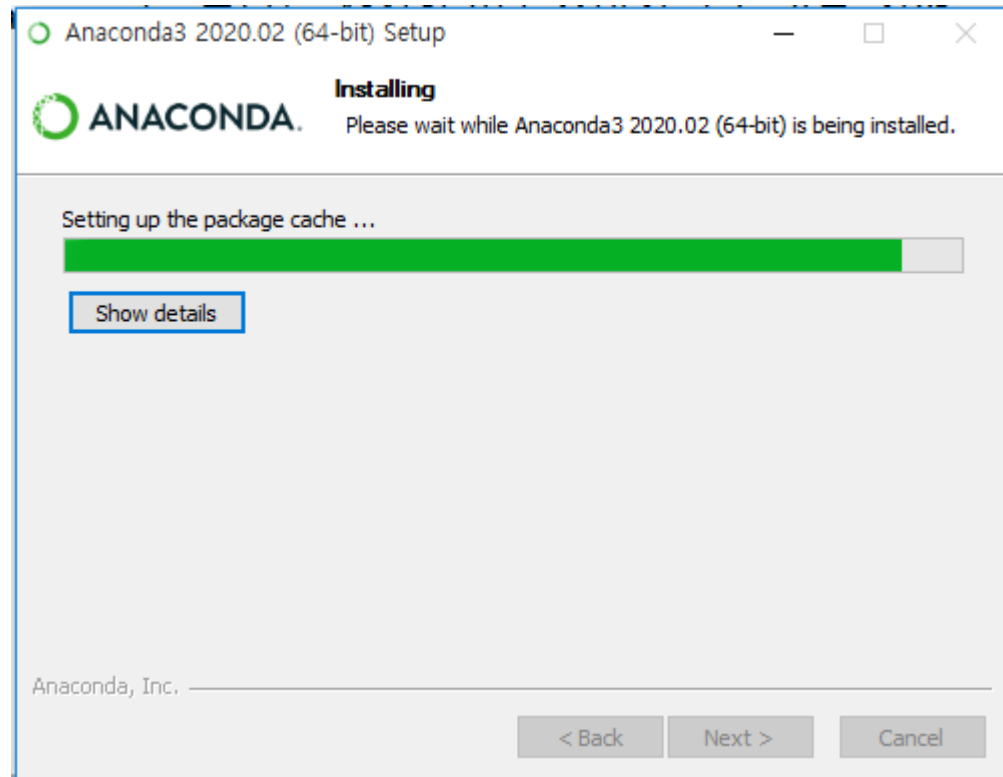
C:\ca\_da\anaconda 폴더를 작성하거나 주어진 default로 Install

C:\Wanaconda3



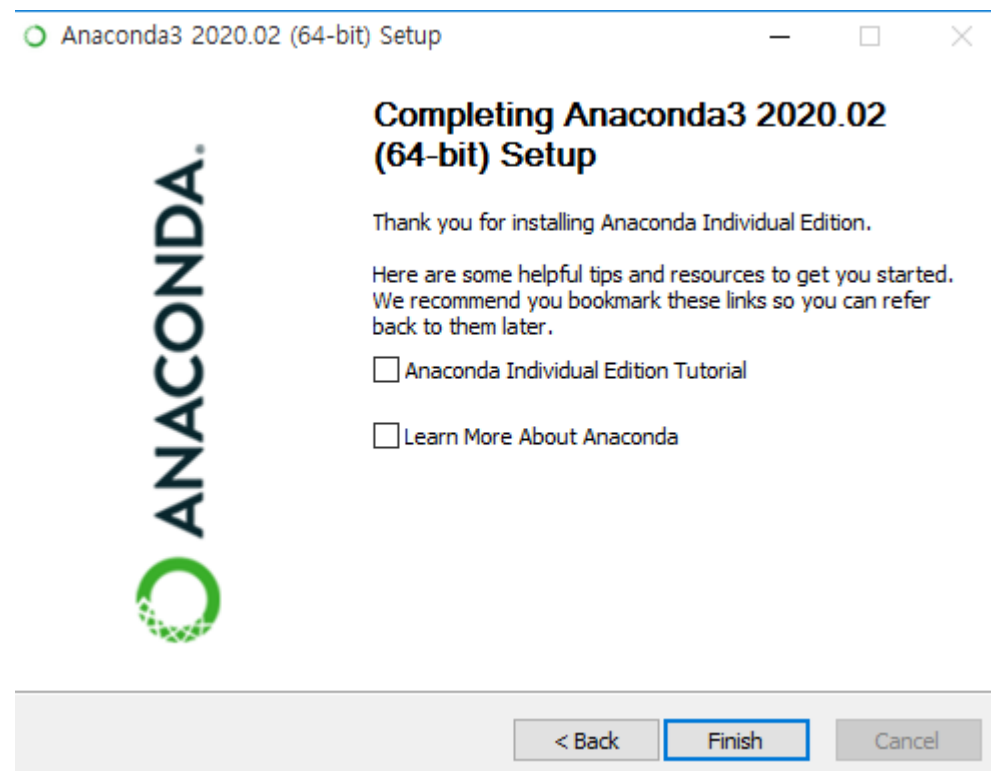
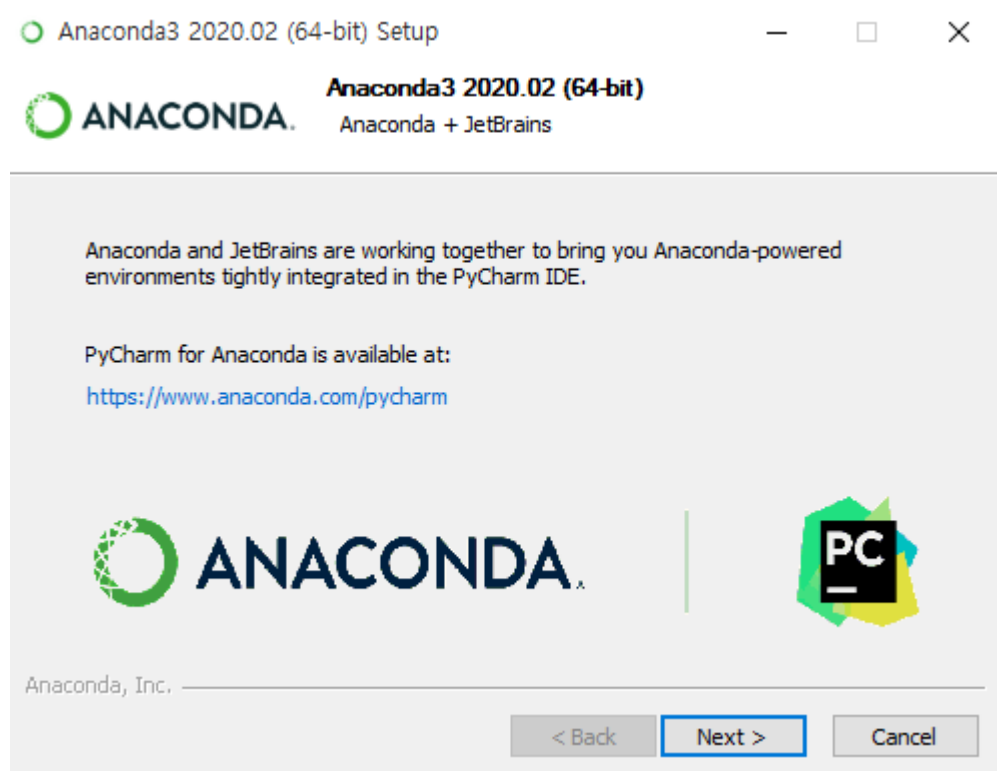
# 아나콘다 설치

Next > Next



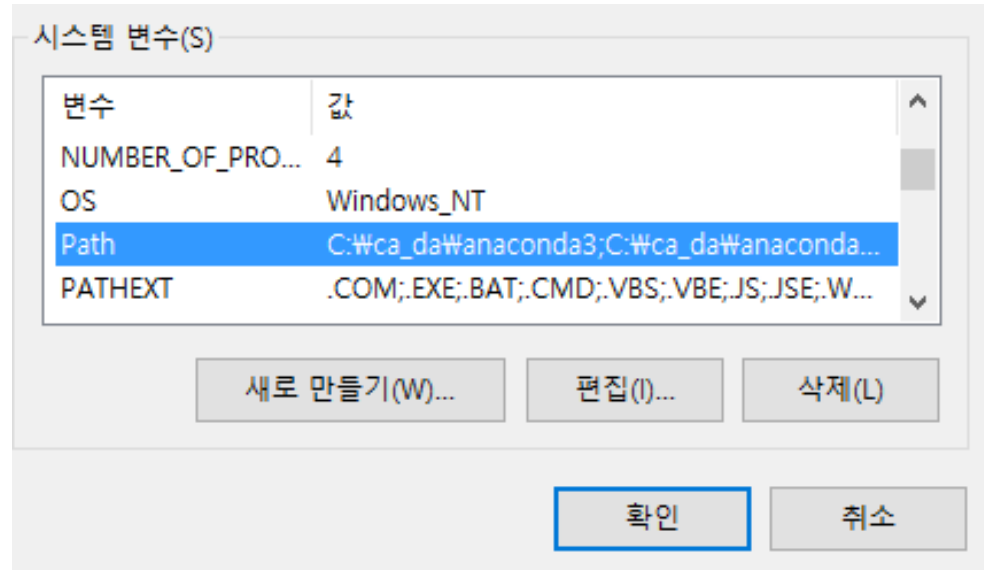
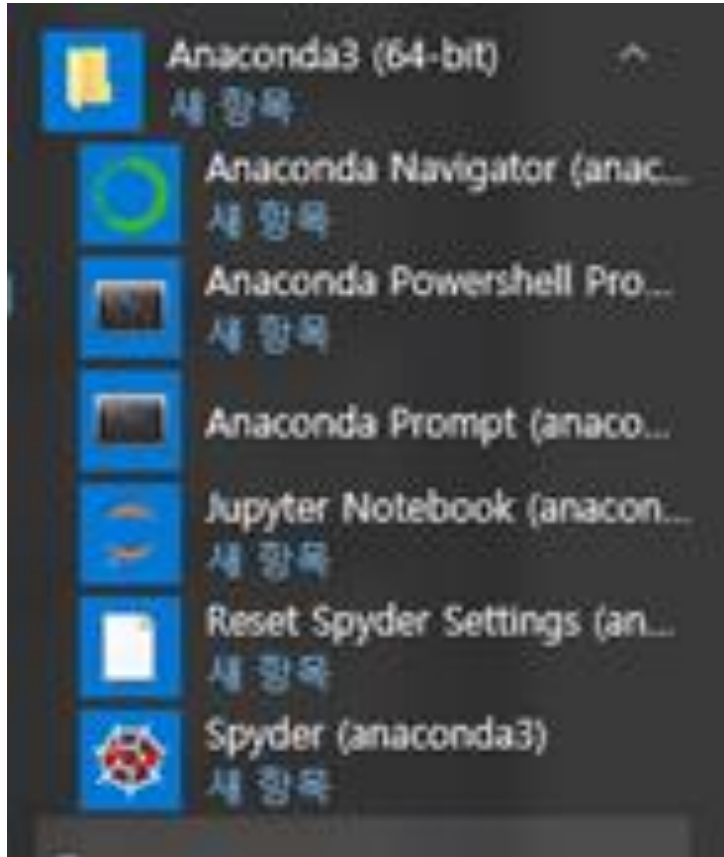
# 아나콘다 설치

Next > Finish



# 아나콘다 설치

설치후 윈도우 시작버튼을 눌러 확인하고 내PC 고급시스템 설정에서 Path 입력



C:\wca\_da\Wanaconda3;  
C:\wca\_da\Wanaconda3\Library\mingw-  
w64\bin;C:\wca\_da\Wanaconda3\Library\bin;  
C:\wca\_da\Wanaconda3\Scripts;

# 아나콘다 활용, 파이썬 가상환경 설치

---

- 버전 명시 설치 : `conda create -n 가상환경명 python=3.7`
- 가상환경 시작 : `activate 가상환경명`
- Jupyter notebook 설치 : `conda install jupyter notebook`
- 패키지 설치 : `conda install numpy pandas matplotlib seaborn  
scipy scikit-learn tensorflow keras`  
# 기본채널에 패키지 부재시 : `conda install -c conda-forge`
- 가상환경 종료 : `conda deactivate`
- 가상환경 저장 : `conda env export -n 가상환경명 > 파일명.yml`
- 새로운 가상환경 생성 : `conda env create -n 가상환경명 -f ./파일명.yml`
- 가상환경 복사 : `conda create -n 생성할가상환경명 --clone 원본가상환경명`
- 가상환경 제거 : `conda env remove -n 가상환경명`



# Anaconda 명령어

---

- 설치된 패키지 : `conda list`
- 패키지 업데이트 : `conda update pandas`, `conda update -all`
- 패키지 제거 : `conda remove pandas`
- 설치된 패키지 검색 : `conda search pandas`
- 가상환경 시작 : `activate 가상환경명`
- 가상환경 종료 : `conda deactivate`

# Anaconda 명령어

---

- `conda install [package name](=[version])`
- ※ `conda install` 은 `anaconda`에서 관리하는 패키지를 `conda` 가상환경에 설치, `pip`는 파이썬 패키지 설치
- ※ `anaconda` 기본 `repo`에 없는 패키지일 경우 `-c [channel]` 옵션을 붙여서 `repository` 설정 가능
- `conda install -c conda-forge 패키지명`  
-c 채널 옵션에 `conda-forge`를 주어서 패키지를 다운로드한다.  
`Conda-forge` 는 `anaconda`에서 쉽게 설치할 수 있도록 검증된 파이썬 패키지들을 모아 놓은 하나의 채널 이다.

# Anaconda 명령어

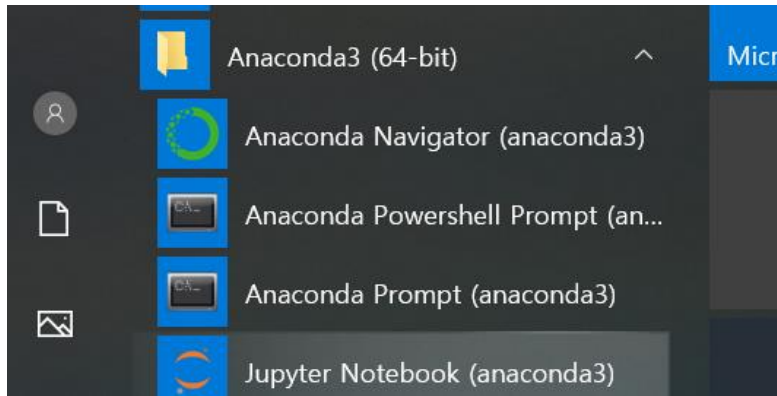
---

## Options:

- **clean** : Remove unused packages and caches.
- **create** : Create a new conda environment from a list of specified packages.
- **help** : Displays a list of available conda commands and their help strings.
- **info** : Display information about current conda install.
- **init** : Initialize conda for shell interaction. [Experimental]
- **remove** : Remove a list of packages from a specified conda environment.
- **uninstall** : Alias for conda remove.
- **search** : Search for packages and display associated information. The input is a MatchSpec, a query language for conda packages.
- **update** : Updates conda packages to the latest compatible version.
- **upgrade** : Alias for conda update.

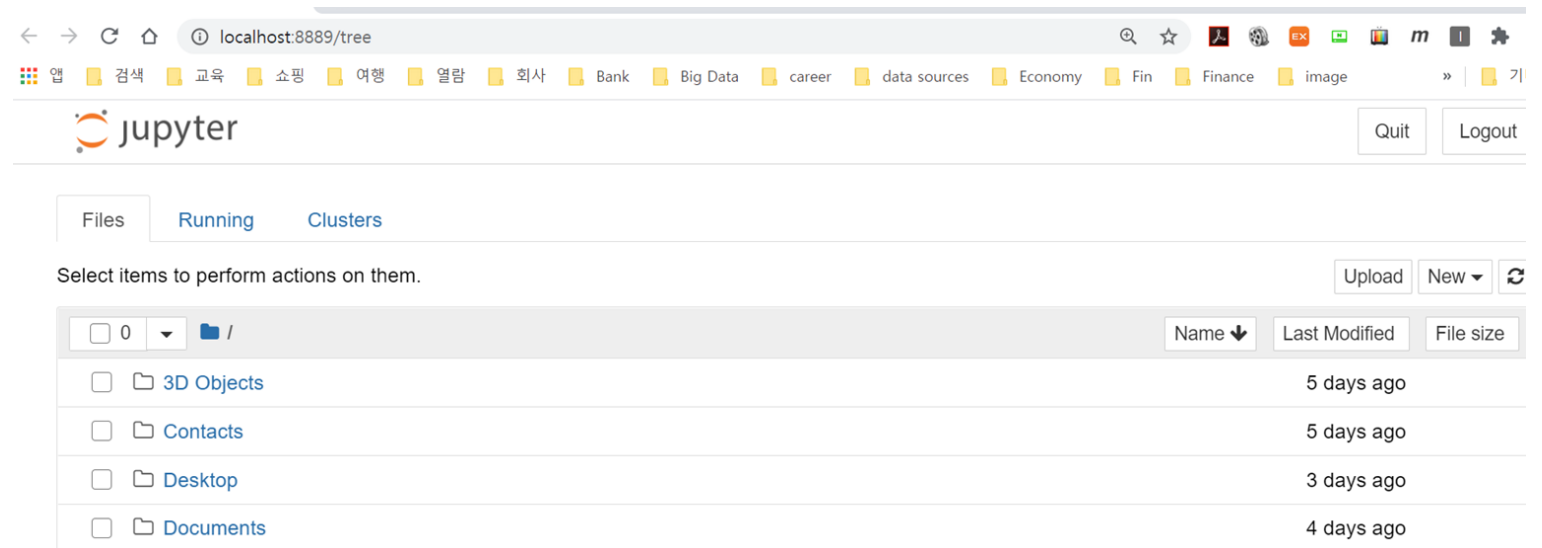
# Jupyter notebook 실행

설치후 윈도우 시작버튼  
을 눌러 Anaconda3에  
있는 Jupyter Notebook  
을 클릭



```
Jupyter Notebook (anaconda3)
[11:40:25.209 NotebookApp] The port 8888 is already in use, trying another port.
[11:40:25.490 NotebookApp] JupyterLab extension loaded from C:\Wanaconda3\lib\site-packages\jupyterlab
[11:40:25.490 NotebookApp] JupyterLab application directory is C:\Wanaconda3\share\jupyterlab
[11:40:25.490 NotebookApp] Serving notebooks from local directory: C:\Users\Wkevin
[11:40:25.490 NotebookApp] The Jupyter Notebook is running at:
[11:40:25.490 NotebookApp] http://localhost:8889/?token=f3cc74f2710e82ca46161a8787533bd1434b8dbd37cdc239
[11:40:25.490 NotebookApp] or http://127.0.0.1:8889/?token=f3cc74f2710e82ca46161a8787533bd1434b8dbd37cdc239
[11:40:25.490 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:40:25.521 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/kevin/AppData/Roaming/jupyter/runtime/nbserver-14528-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=f3cc74f2710e82ca46161a8787533bd1434b8dbd37cdc239
or http://127.0.0.1:8889/?token=f3cc74f2710e82ca46161a8787533bd1434b8dbd37cdc239
```



# Jupyter notebook – Themes 적용

---

**command prompt 또는 anaconda prompt에서 □□ 설치**

**pip install jupyterthemes**

**jt -l # 적용 가능한 테마 목록 출력**

**jt -t grade3 # grade3 테마 적용**

**jt -t grade3 -f bitstream -T -N -fs 12 -nf opensans -nfs 10**

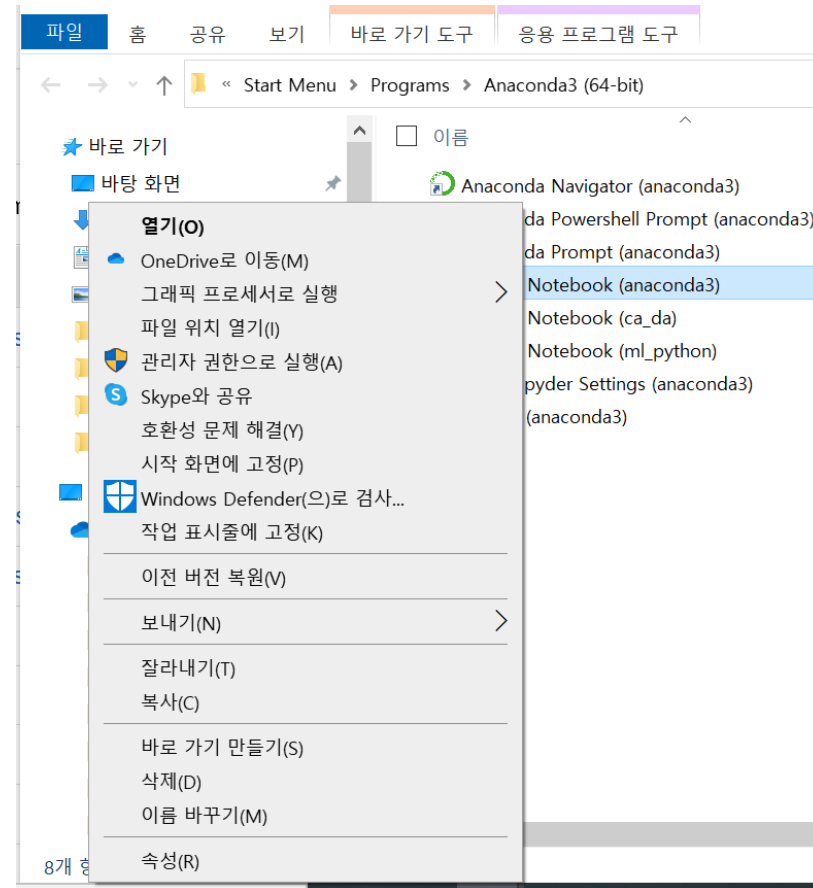
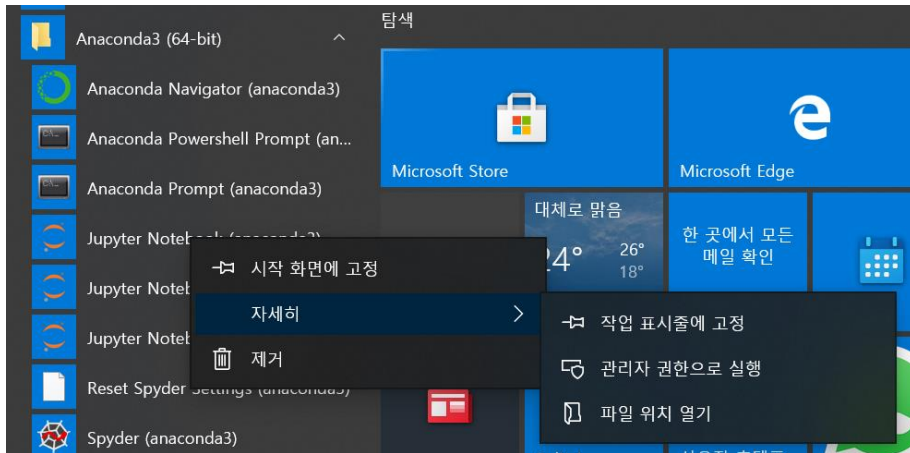
- -f bitstream : 코드 폰트를 bitstream으로 적용
- -fs 12 : 글자 크기 12
- -T -N : 툴바와 제목을 보임 // default는 안보이는 상태라서 코드를 더 많이 볼 수 있습니다.
- -nf : 주피터노트북 폰트를 opensans로 적용
- -nfs 10 : 노트북 글자 크기 10

**다시 원래대로 돌아가기**

**Jt -r**

# Jupyter notebook – 작업폴더 지정

시작버튼 Anaconda3 > Jupyter notebook 우클릭 > 자세히 > 파일 위치 열기  
Jupyter notebook(anaconda3) 파일 우클릭 > 속성 클릭



# Windows Ubuntu 터미널에 Anaconda를 설치하는 단계

<https://www.emilykauffman.com/teaching/install-anaconda-on-wsl>

- wget [https://repo.continuum.io/archive/Anaconda3-5.2.0-Linux-x86\\_64.sh](https://repo.continuum.io/archive/Anaconda3-5.2.0-Linux-x86_64.sh)
- 설치 스크립트 실행 : `bash Anaconda3-5.2.0-Linux-x86_64.sh`
- 선택적으로 메시지가 표시되면 VS Code를 설치
- 터미널을 닫고 다시 열어 .bash 구성을 다시 로드
- `which python` : anaconda가 있는 경로 “/home/kevin/anaconda3/bin/python” 확인
- anaconda가 없으면 다음 단계를 수행  
수동으로 anaconda bin 폴더를 PATH에 추가. 이를 위해 ~/.bashrc 파일의 맨 아래에  
"export PATH=/home/kevin/anaconda3/bin:\$PATH"를 추가
- `jupyter notebook --no-browser`
- C:/Users/kevin/Documents/JupyterNotebooks폴더(Windows에서 다운로드한 노트북을 저장하는 위치)와 Ubuntu 노트북 디렉토리 사이에 심볼릭 링크 생성  
`cd ~`  
`ln -s /mnt/c/Users/kevin/Documents/JupyterNotebooks/ notebooks`
- alias jup 등록 : `vim ~/.bashrc` > `alias jup='cd /home/kevin/notebooks && jupyter notebook --no-browser'` > `source ~/.bashrc`

# Ipython 및 주피터 설정 디렉토리

---

- 프로필 작성(위치 : 사용자홈디렉토리/.ipython/profile\_default/  
프로필 디렉토리가 없다면 ipython profile create로 만들어줌
- 사용자 설정 파일 : 스타트업 파일, ipython\_config.py 파일  
touch ~/.ipython/profile\_default/startup/00.py
- vim ~/.ipython/profile\_default/startup/00.py  
[https://raw.githubusercontent.com/datasciencesschool/docker\\_rpython/master/02\\_rpython/00.py](https://raw.githubusercontent.com/datasciencesschool/docker_rpython/master/02_rpython/00.py)
- vim ~/.ipython/profile\_default/ipython\_config.py  
[https://raw.githubusercontent.com/datasciencesschool/docker\\_rpython/master/02\\_rpython/ipython\\_config.py](https://raw.githubusercontent.com/datasciencesschool/docker_rpython/master/02_rpython/ipython_config.py)
- sudo apt-get install -y fonts-nanum\*
- ipython > import matplotlib > [f.name for f in matplotlib.font\_manager.fontManager.ttflist if f.name.startswith('N')]



# PyTorch 설치 및 구성

---

<https://docs.microsoft.com/ko-kr/windows/ai/windows-ml/tutorials/pytorch-analysis-installation>  
- conda install pytorch torchvision torchaudio cpuonly -c pytorch

# WSL2 설치와 VSCode 연동하기

<https://velog.io/@gidskql6671/WSL-WSL2-%EC%84%A4%EC%B9%98-VSCode-%EC%97%B0%EB%8F%99>

- Terminal에서 원하는 프로젝트 디렉토리로 이동한 다음 code .
- VSCode Extension에서 Remote - WSL을 설치
- WSL환경이 아닌 Windows 환경에서 VSCode를 이용하고 싶은 경우 좌측 하단에 초록색 아이콘을 클릭, 원격 연결 단기를 통해 WSL과의 연결을 해제
- WSL과의 연결을 해제한 경우, 최근에 연결했던 프로젝트들을 쉽게 연결할 수 있는 방법 제공  
VSCode 좌측 네비게이션 바에서 6번째 컴퓨터모양 아이콘을 클릭하면 이전에 연결해 두었던 연결들이 남아있는 것을 볼 수 있으며 첫번째가 Open Folder in WSL으로 해당 버튼으로 다시 연결. 두번째 버튼은 Remove Recent Folder로 해당 프로젝트를 목록에서 삭제
- WSL은 Windows와 다른 별개의 환경이고, Vscod은 원격으로 해당 환경에 접속하는 것으로 Extension 또한 로컬과 별개로 다르게 설치해줄 수 있다.