

Python 설치 및 환경설정

양승우

먼저 이 책의 예제를 사용하려면 깃허브(Git-Hub) 저장소를 클론해야 합니다.

깃(Git)은 오픈소스 버전 관리 시스템입니다.

주피터 노트북(jupyter notebook)을 실행 시켰을 때,

git clone을 했다면 GDL_code 파일에 모든 예제가 있습니다.

https://github.com/rickiepark/GDL_code 참고하세요.



1. Download x.xx for Windows 클릭



The screenshot shows the Git website homepage. At the top, the Git logo is followed by the tagline "--fast-version-control". A search bar is located in the top right corner. The main content area features a description of Git as a "free and open source distributed version control system" and another paragraph highlighting its "easy to learn" nature and "tiny footprint with lightning fast performance". To the right of the text is a diagram illustrating Git's branching model with stacks of code and colored lines representing branches. Below the main text, there are four circular icons with corresponding text: "About" (gears), "Documentation" (book), "Downloads" (downward arrow), and "Community" (speech bubbles). On the right side, a monitor displays the "Latest source Release 2.27.0" with a link to "Release Notes (2020-06-01)" and a button labeled "Download 2.27.0 for Windows" which is highlighted with a red rectangle.

git --fast-version-control

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.27.0
[Release Notes \(2020-06-01\)](#)
Download 2.27.0 for Windows

<https://git-scm.com/>

2. 컴퓨터에 알맞은 버전(version) 선택 (대부분 자동으로 맞는 버전이 다운로드 됩니다.)

지금은 설치만 해 두고 6번에서 다시 쓰겠습니다.

Downloading Git



Your download is starting...

You are downloading the latest (2.27.0) 64-bit version of Git for Windows. This is the most recent [maintained build](#). It was released 30 days ago, on 2020-06-01.

[Click here to download manually](#), if your download hasn't started.

Other Git for Windows downloads

Git for Windows Setup

[32-bit Git for Windows Setup.](#)

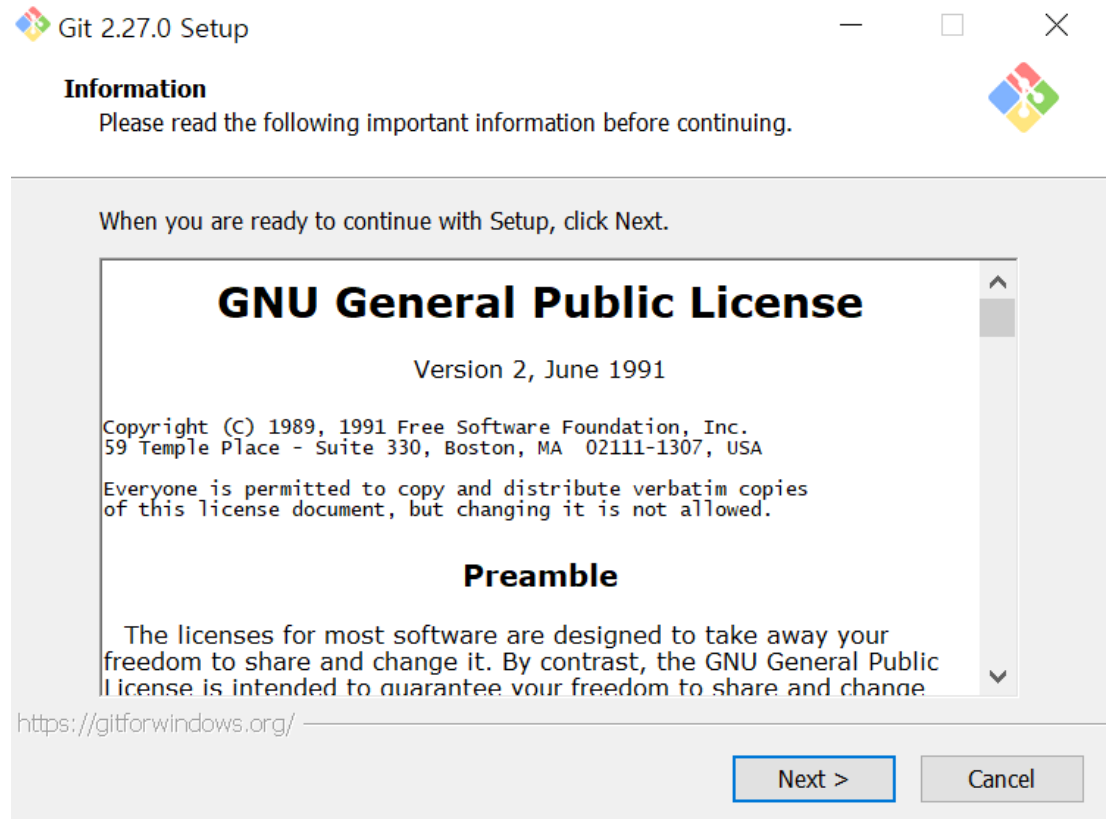
[64-bit Git for Windows Setup.](#)

Git for Windows Portable ("thumbdrive edition")

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

The current source code release is version 2.27.0. If you want the newer version, you can build it from [the source code](#).



* 설치 시에 특별한 설정 없이 Next 버튼을 눌러 설치하면 됩니다.

이제 파이썬(Python)이라는

프로그래밍 언어로 프로그래밍을 공부하기에 앞서 먼저 파이썬을 설치해 보겠습니다.

파이썬을 설치하려면 먼저 설치 프로그램을 내려받아야 합니다.

파이썬 설치 파일은 크게 두 가지로 구분됩니다.

- 파이썬 공식 홈페이지에서 제공하는 설치 파일
- 배포판이라고 불리는 설치 파일



첫 번째는 파이썬 공식 홈페이지(<http://www.python.org>)에서 제공하는 설치 파일로서 C-Python이라고 합니다.

해당 파일을 내려받아 설치하면 파이썬 인터프리터와 몇 가지 프로그램이 설치됩니다.
이 설치 파일은 파이썬으로 프로그래밍하는데 필요한 최소한의 프로그램만 컴퓨터에 설치합니다.
그래서 설치 파일의 용량이 24MB 정도로 적은 편입니다.

두 번째는 배포판이라고 불리는 설치 파일입니다.

뒤에서 설명하겠지만 파이썬에는 파이썬 패키지라고 하는 것들이 있는데,
여기서 패키지라는 것은 특정 분야에 최적화된 프로그램의 묶음이라고 보면 됩니다.

예를 들어, 주식 프로그램을 작성하는데 처음부터 모든 것을 다 작성하는 것이 아니라
주식 패키지 를 설치하면 기본적인 것들이 이미 구현돼 있는 것이죠.

어떤가요? 말로만 들어도 편하겠지요?
좀 더 쉽게 설명하면 어떤 보고서를 작성하는데,
이미 누군가가 그 보고서의 초안을 작성한 후 제공하는 것과 비슷합니다.



그런데 문제는 파이썬에는 패키지가 너무나도 많아서
필요한 패키지를 그때그때 내려받아서 직접 설치해야 한다는 것입니다.

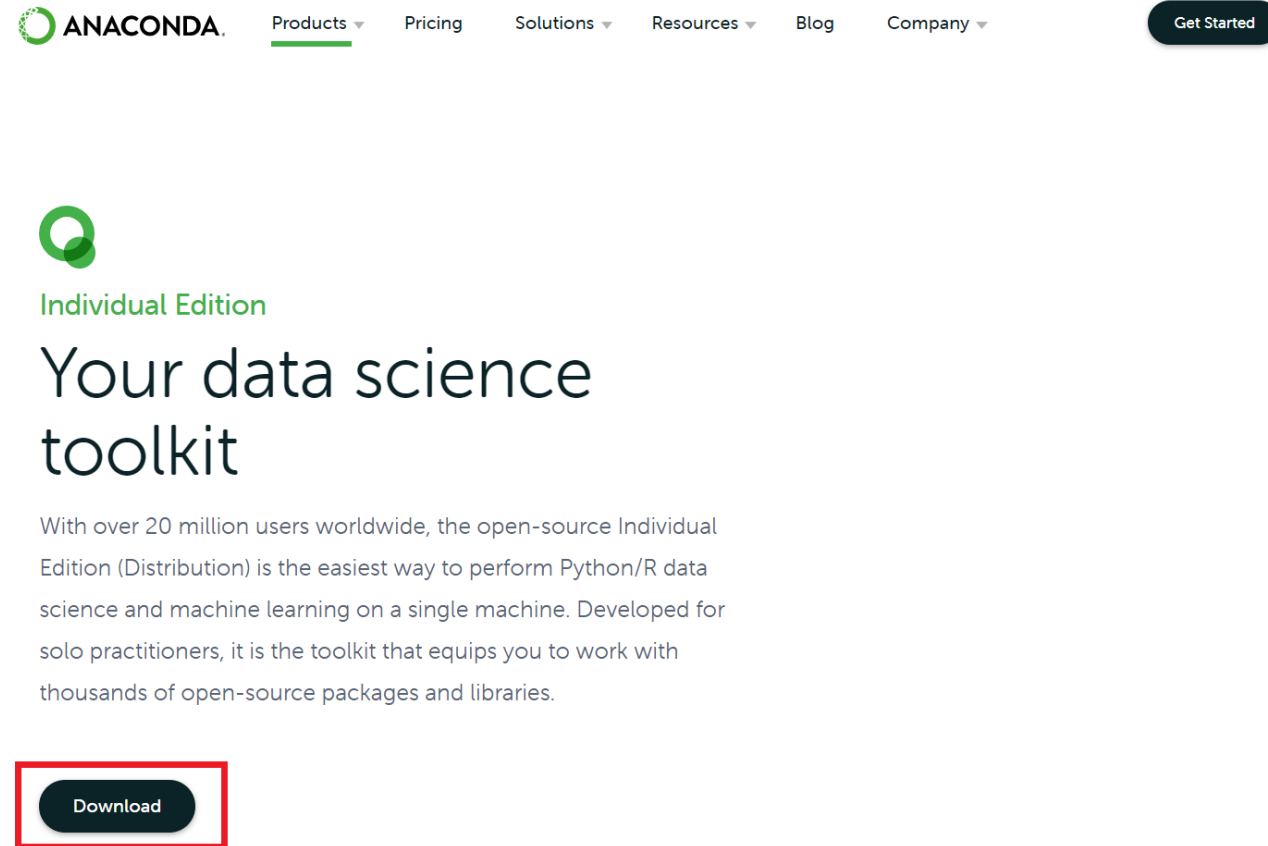
이러한 문제를 깔끔하게 해결하고자 배포판이라고 하는 설치 파일들이 등장했습니다.

쉽게 말해서 배포판은 '파이썬 인터프리터 + 파이썬 패키지'라고 생각하면 됩니다.
배포판은 파이썬 인터프리터와 자주 사용하는 백여 개의 패키지를 한 번에 컴퓨터에 설치해줍니다.
다만 설치 파일의 크기가 200MB~1GB 정도로 크다는 단점이 있습니다.

물론 둘 중 어떤 설치 파일로 설치하든 파이썬 프로그램을 작성하는 데는 문제가 없지만,
여기서는 여러 유용한 패키지까지 한 번에 설치할 수 있는
'아나콘다'라는 배포판을 이용할 것입니다.




1. Download 클릭



The screenshot shows the top section of the Anaconda Individual Edition website. At the top is the Anaconda logo and a navigation menu with links for Products, Pricing, Solutions, Resources, Blog, and Company. A 'Get Started' button is located in the top right corner. Below the navigation bar, the Anaconda logo is displayed again, followed by the text 'Individual Edition'. The main heading reads 'Your data science toolkit'. A paragraph of text describes the product as an open-source toolkit for Python/R data science and machine learning. At the bottom of this section, a 'Download' button is highlighted with a red rectangular border.

ANACONDA. Products Pricing Solutions Resources Blog Company [Get Started](#)

 Individual Edition

Your data science toolkit

With over 20 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.

[Download](#)

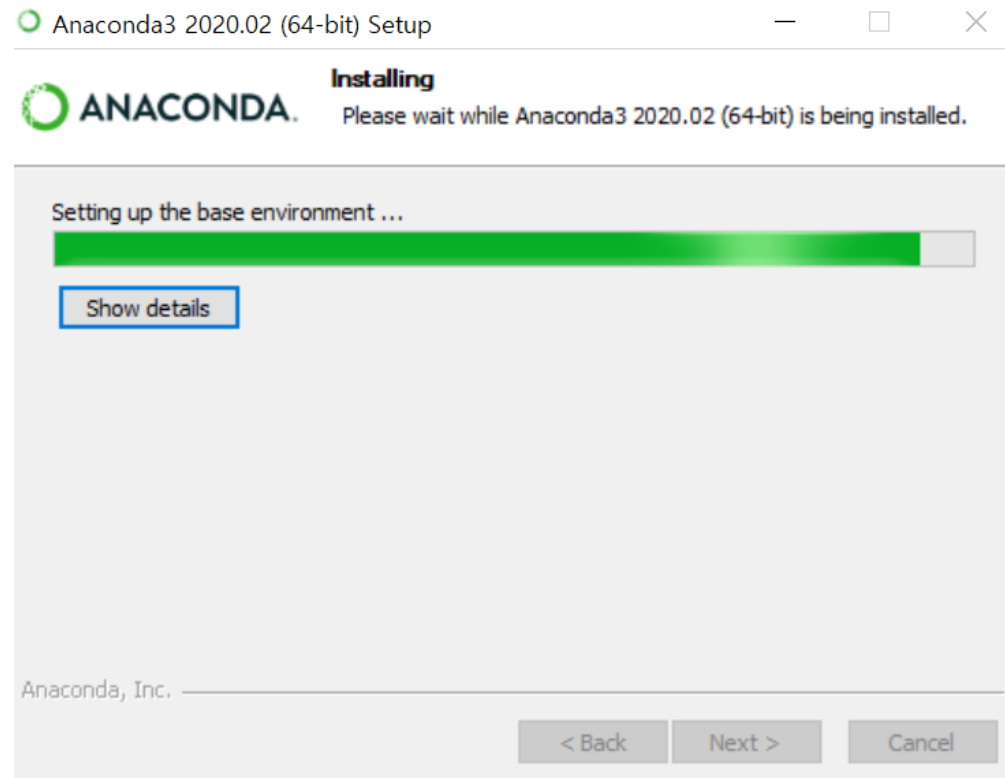
<https://www.anaconda.com/distribution>

2. 컴퓨터에 알맞은 버전(version) 선택

Anaconda Installers

Windows 	MacOS 	Linux 
Python 3.7 64-Bit Graphical Installer (466 MB) 32-Bit Graphical Installer (423 MB)	Python 3.7 64-Bit Graphical Installer (442 MB) 64-Bit Command Line Installer (430 MB)	Python 3.7 64-Bit (x86) Installer (522 MB) 64-Bit (Power8 and Power9) Installer (276 MB)
Python 2.7 64-Bit Graphical Installer (413 MB) 32-Bit Graphical Installer (356 MB)	Python 2.7 64-Bit Graphical Installer (637 MB) 64-Bit Command Line Installer (409 MB)	Python 2.7 64-Bit (x86) Installer (477 MB) 64-Bit (Power8 and Power9) Installer (295 MB)

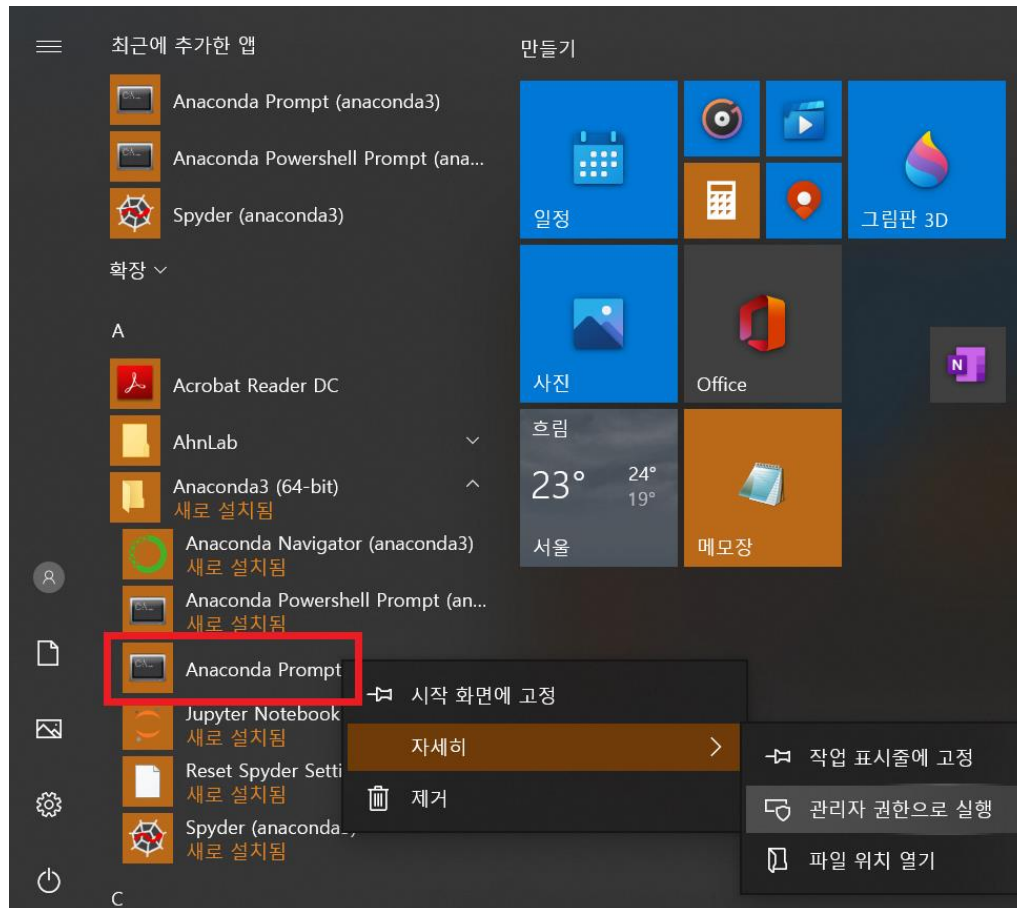
설치 진행 중...



* 설치시에 특별한 설정 없이 Next 버튼을 눌러 설치하면 됩니다. (5분 가량 소요)

3. 안전성과 호환성의 이유로 설치된 최신버전이 아닌 지금의 환경과 다른 가상의 환경을 만듭니다.

가상의 버전을 만들기 위해 Anaconda Prompt를 관리자 권한으로 실행해 줍니다.



관리자: Anaconda Prompt (anaconda3)

```
(base) C:\WINDOWS\system32>python --version  
Python 3.7.6  
  
(base) C:\WINDOWS\system32>
```

* python --version 명령어로 현재 버전 확인

4. 다음과 같이 명령어를 입력합니다. (중요)
초록색 네모칸은 가상환경의 이름입니다. 이름 설정은 자유입니다.

■ 관리자: Anaconda Prompt (anaconda3)

```
(base) C:\WINDOWS\system32>python --version  
Python 3.7.6
```

```
(base) C:\WINDOWS\system32>conda create -n testGAN python=3.6 anaconda
```

* conda create -n (가상환경 이름) python=3.6 anaconda

(python 3.6 anaconda 패키지 버전으로 가상 환경을 생성합니다.)

설치 진행 중... (20분 정도 소요)

```
관리자: Anaconda Prompt (anaconda3) - conda create -n testGAN python=3.6 anaconda
wheel pkgs/main/win-64::wheel-0.34.2-py36_0
widgetsnbextension pkgs/main/win-64::widgetsnbextension-3.5.1-py36_0
win_inet_pton pkgs/main/win-64::win_inet_pton-1.1.0-py36_0
win_unicode_console pkgs/main/win-64::win_unicode_console-0.5-py36hcd4b5_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py36h7fe50ca_0
winpty pkgs/main/win-64::winpty-0.4.3-4
wrapit pkgs/main/win-64::wrapit-1.12.1-py36he774522_1
xlrd pkgs/main/win-64::xlrd-1.2.0-py36_0
xlsxwriter pkgs/main/noarch::xlsxwriter-1.2.9-py_0
xlwings pkgs/main/win-64::xlwings-0.19.4-py36_0
xlwt pkgs/main/win-64::xlwt-1.3.0-py36h1a4751e_0
xz pkgs/main/win-64::xz-5.2.5-h62dcd97_0
yaml pkgs/main/win-64::yaml-0.2.5-he774522_0
yapf pkgs/main/noarch::yapf-0.30.0-py_0
zeromq pkgs/main/win-64::zeromq-4.3.2-ha925a31_2
zict pkgs/main/noarch::zict-2.0.0-py_0
zipp pkgs/main/noarch::zipp-3.1.0-py_0
zlib pkgs/main/win-64::zlib-1.2.11-h62dcd97_4
zope pkgs/main/win-64::zope-1.0-py36_1
zope.event pkgs/main/win-64::zope.event-4.4-py36_0
zope.interface pkgs/main/win-64::zope.interface-4.7.1-py36he774522_0
zstd pkgs/main/win-64::zstd-1.4.4-ha9fde0e_3

Proceed ([y]/n)? y

Downloading and Extracting Packages
matplotlib-base-3.2.0 | 5.4 MB | ##### | 100%
certifi-2020.6.20 | 156 KB | ##### | 100%
yaml-0.2.5 | 62 KB | ##### | 100%
pylint-2.4.4 | 445 KB | ##### | 100%
pyopenssl-19.1.0 | 91 KB | ##### | 100%
decorator-4.4.2 | 14 KB | ##### | 100%
pexpect-4.8.0 | 83 KB | ##### | 100%
wheel-0.34.2 | 66 KB | ##### | 100%
py-1.9.0 | 79 KB | ##### | 100%
kiwisolver-1.2.0 | 55 KB | ##### | 100%
astroid-2.3.3 | 282 KB | ##### | 100%
jupyterlab-2.1.5 | 3.3 MB | ##### | 100%
more-iterutils-8.4.0 | 42 KB | ##### | 100%
```

* y 를 눌러 계속 진행합니다.

설치 진행 중... (20분 정도 소요)

```
관리자: Anaconda Prompt (anaconda3)
- DEBUG menuinst_win32: __init__(199): Menu: name: 'Anaconda${PY_VER} ${PLATFORM}', prefix: 'C:\Users\User\anaconda3\envs\testGAN', env_name: 'testGAN', mode:
  'user', used_mode: 'user'
DEBUG menuinst_win32: create(323): Shortcut cmd is %windir%\System32\cmd.exe, args are ['"/K"', 'C:\Users\User\anaconda3\Scripts\activate.bat', 'C:\Users\User\anaconda3\envs\testGAN']
- DEBUG menuinst_win32: __init__(199): Menu: name: 'Anaconda${PY_VER} ${PLATFORM}', prefix: 'C:\Users\User\anaconda3\envs\testGAN', env_name: 'testGAN', mode:
  'user', used_mode: 'user'
DEBUG menuinst_win32: create(323): Shortcut cmd is %windir%\System32\WindowsPowerShell\v1.0\powershell.exe, args are ['-ExecutionPolicy', 'Bypass', '-NoExit',
'-Command', '& C:\Users\User\anaconda3\shell\condabin\conda-hook.ps1 & conda activate C:\Users\User\anaconda3\envs\testGAN']
DEBUG menuinst_win32: __init__(199): Menu: name: 'Anaconda${PY_VER} ${PLATFORM}', prefix: 'C:\Users\User\anaconda3\envs\testGAN', env_name: 'testGAN', mode:
  'user', used_mode: 'user'
DEBUG menuinst_win32: create(323): Shortcut cmd is C:\Users\User\anaconda3\python.exe, args are ['C:\Users\User\anaconda3\cwp.py', 'C:\Users\User\anaconda3\envs\testGAN', 'C:\Users\User\anaconda3\envs\testGAN\python.exe', 'C:\Users\User\anaconda3\envs\testGAN\Scripts\jupyter-notebook-script.py', '%USERPROFILE%\']
- DEBUG menuinst_win32: __init__(199): Menu: name: 'Anaconda${PY_VER} ${PLATFORM}', prefix: 'C:\Users\User\anaconda3\envs\testGAN', env_name: 'testGAN', mode:
  'user', used_mode: 'user'
DEBUG menuinst_win32: create(323): Shortcut cmd is C:\Users\User\anaconda3\pythonw.exe, args are ['C:\Users\User\anaconda3\cwp.py', 'C:\Users\User\anaconda3\envs\testGAN', 'C:\Users\User\anaconda3\envs\testGAN\pythonw.exe', 'C:\Users\User\anaconda3\envs\testGAN\Scripts\spyder-script.py']
DEBUG menuinst_win32: create(323): Shortcut cmd is C:\Users\User\anaconda3\python.exe, args are ['C:\Users\User\anaconda3\cwp.py', 'C:\Users\User\anaconda3\envs\testGAN', 'C:\Users\User\anaconda3\envs\testGAN\python.exe', 'C:\Users\User\anaconda3\envs\testGAN\Scripts\spyder-script.py', '--reset']
done
#
# To activate this environment, use
#
#     $ conda activate testGAN
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) C:\WINDOWS\system32>
```

5. 생성된 가상환경을 활성화 명령어로 활성화 시키고 가상환경의 버전을 확인합니다.

활성화 되었다면 (base) → (설정한 가상환경 이름)으로 바뀝니다. (초록색 부분)

버전이 python 3.6.xx로 변경되었는지 확인합니다.

관리자: Anaconda Prompt (anaconda3)

```
(base) C:\WINDOWS\system32>activate testGAN
```

```
(testGAN) C:\WINDOWS\system32>python --version  
Python 3.6.10 :: Anaconda, Inc.
```

* activate (설정한 가상환경 이름)

6. 복사해 올 경로를 설정합니다. 조금 전에 설치한 git clone을 사용합니다.

■ 관리자: Anaconda Prompt (anaconda3)

```
(base) C:\WINDOWS\system32>activate testGAN

(testGAN) C:\WINDOWS\system32>python --version
Python 3.6.10 :: Anaconda, Inc.

(testGAN) C:\WINDOWS\system32>cd C:\Users\User

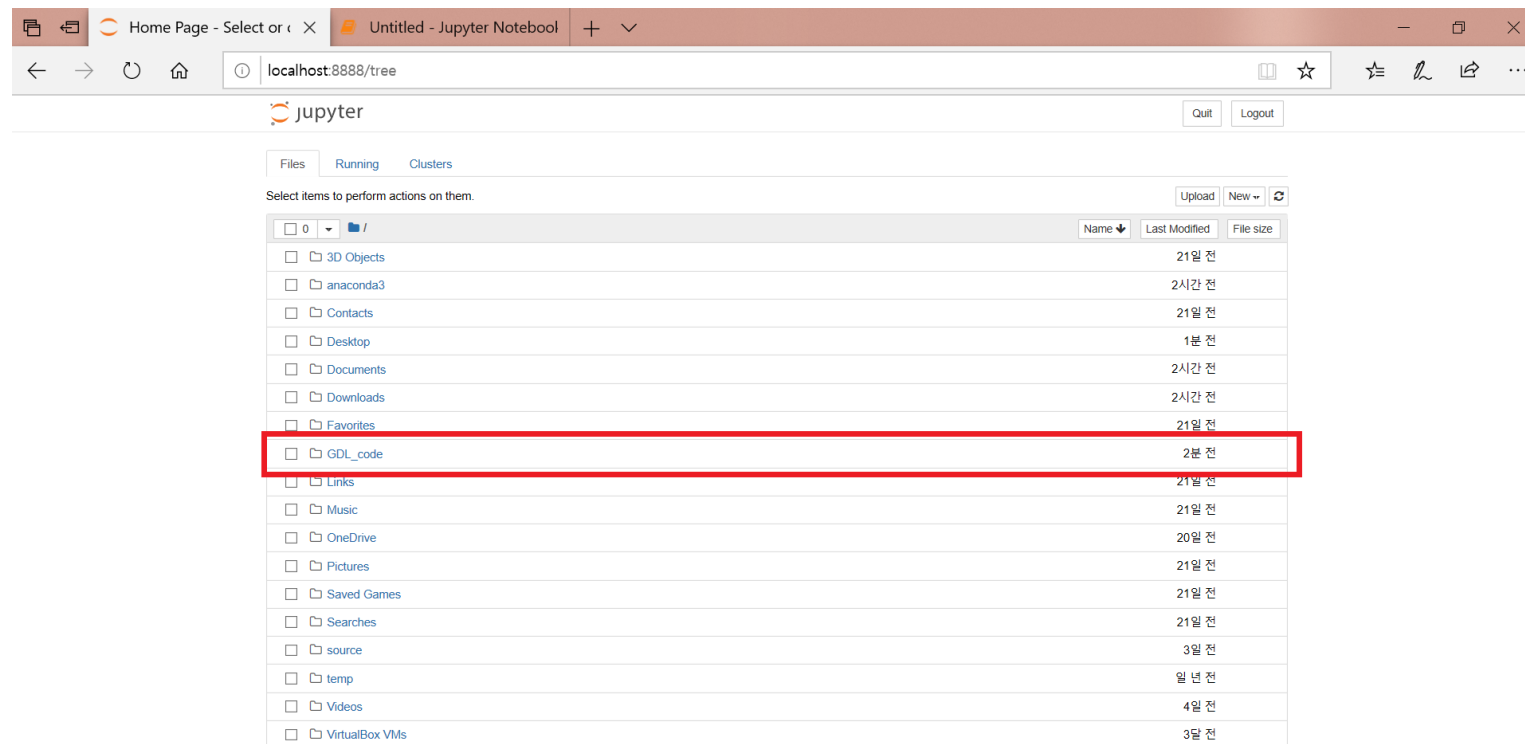
(testGAN) C:\Users\User>git clone https://github.com/rickiepark/GDL_code.git
Cloning into 'GDL_code' ...
remote: Enumerating objects: 480, done.
remote: Total 480 (delta 0), reused 0 (delta 0), pack-reused 480
Receiving objects: 100% (480/480), 178.44 MiB | 17.11 MiB/s, done.
Resolving deltas: 100% (224/224), done.
Updating files: 100% (77/77), done.
```

* cd C:\Users\User (경로 설정 명령어)

* git clone https://github.com/rickiepark/GDL_code.git (복사 명령어)

모든 설치 후에 Jupyter notebook에 접속해 보면 GDL_code파일이 생깁니다.

여기에 모든 예제가 있습니다.



딥러닝(Deep Learning)이란 ?

머신 러닝(Machine Learning)중에서도 인공 신경망을 이용해 학습을 하는 러닝의 종류를 의미한다.
이를 위한 라이브러리가 많이 있는데, 파이썬에서는 텐서플로우(Tensorflow) 또는 테아노(Theano)같은
오픈소스 라이브러리들이 존재한다.

하지만 이 라이브러리들은 파라미터나 설정 등이 너무 복잡해서 사용하기 어렵다.
그래서 사람들이 Keras라는 라이브러리를 만들어 쉽게 인공신경망을 만들 수 있게 했다.
물론 나중에 fine-tuning을 하기 위해서는 결국 텐서플로우(TensorFlow)를 직접 다뤄야 할 지 모르지만,
여기서는 Keras를 주로 사용한다.



5. 다음 명령어로 keras 2.2.4버전 설치합니다.

반드시 설정한 가상환경 이름으로 activate 된 상태로 설치해야 합니다.

관리자: Anaconda Prompt (anaconda3)

```
(base) C:\WINDOWS\system32>activate testGAN
(testGAN) C:\WINDOWS\system32>python --version
Python 3.6.10 :: Anaconda, Inc.
(testGAN) C:\WINDOWS\system32>pip install keras==2.2.4
Collecting keras==2.2.4
  Using cached Keras-2.2.4-py2.py3-none-any.whl (312 kB)
Requirement already satisfied: h5py in c:\users\user\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (2.10.0)
Requirement already satisfied: numpy>=1.9.1 in c:\users\user\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (1.18.5)
Requirement already satisfied: pyyaml in c:\users\user\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (5.3.1)
Requirement already satisfied: six>=1.9.0 in c:\users\user\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (1.15.0)
Collecting keras-applications>=1.0.6
  Using cached Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
Requirement already satisfied: scipy>=0.14 in c:\users\user\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (1.5.0)
Collecting keras-preprocessing>=1.0.5
  Using cached Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Installing collected packages: keras-applications, keras-preprocessing, keras
Successfully installed keras-2.2.4 keras-applications-1.0.8 keras-preprocessing-1.1.2
```

* pip install keras==2.2.4 (keras 설치 명령어)

5. 다음 명령어로 tensorflow 1.13.2버전 설치합니다.

반드시 설정한 이름으로 activate 된 상태로 설치해야 합니다.

```
관리자: Anaconda Prompt (anaconda3) - pip install tensorflow==1.13.2

(base) C:\WINDOWS\system32>activate testGAN

(testGAN) C:\WINDOWS\system32>python --version
Python 3.6.10 :: Anaconda, Inc.

(testGAN) C:\WINDOWS\system32>pip install keras==2.2.4
Collecting keras==2.2.4
  Using cached Keras-2.2.4-py2.py3-none-any.whl (312 kB)
Requirement already satisfied: h5py in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (2.10.0)
Requirement already satisfied: numpy>=1.9.1 in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (1.18.5)
Requirement already satisfied: pyyaml in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (5.3.1)
Requirement already satisfied: six>=1.9.0 in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (1.15.0)
Collecting keras-applications>=1.0.6
  Using cached Keras_Applications-1.0.8-py3-none-any.whl (50 kB)
Requirement already satisfied: scipy>=0.14 in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from keras==2.2.4) (1.5.0)
Collecting keras-preprocessing>=1.0.5
  Using cached Keras_Preprocessing-1.1.2-py2.py3-none-any.whl (42 kB)
Installing collected packages: keras-applications, keras-preprocessing, keras
Successfully installed keras-2.2.4 keras-applications-1.0.8 keras-preprocessing-1.1.2

(testGAN) C:\WINDOWS\system32>pip install tensorflow==1.13.2
Collecting tensorflow==1.13.2
  Using cached tensorflow-1.13.2-cp36-cp36m-win_amd64.whl (63.1 MB)
Collecting protobuf>=3.6.1
  Using cached protobuf-3.12.2-cp36-cp36m-win_amd64.whl (1.1 MB)
Processing c:\Users\User\appdata\local\pip\cache\wheels\7c\06\54\bc84598ba1daf8f970247f550b175aaee85f68b4b0c5ab2c6\termcolor-1.1.0-cp36-none-any.whl
Collecting grpcio>=1.8.6
  Using cached grpcio-1.30.0-cp36-cp36m-win_amd64.whl (2.4 MB)
Collecting astor>=0.6.0
  Using cached astor-0.8.1-py2.py3-none-any.whl (27 kB)
Collecting tensorboard<1.14.0,>=1.13.0
  Using cached tensorboard-1.13.1-py3-none-any.whl (3.2 MB)
Requirement already satisfied: wheel>=0.26 in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from tensorflow==1.13.2) (0.34.2)
Collecting gast>=0.2.0
  Using cached gast-0.3.3-py2.py3-none-any.whl (9.7 kB)
Processing c:\Users\User\appdata\local\pip\cache\wheels\8e\28\49\fad4e7f0b9a1227708cbbee4487ac8558a7334849cb81c813d\absl_py-0.9.0-cp36-none-any.whl
Requirement already satisfied: numpy<2.0,>=1.16.0 in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from tensorflow==1.13.2) (1.18.5)
Collecting tensorflow-estimator<1.14.0rc0,>=1.13.0
  Using cached tensorflow_estimator-1.13.0-py2.py3-none-any.whl (367 kB)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\Users\User\anaconda3\envs\testgan\lib\site-packages (from tensorflow==1.13.2) (1.1.2)
```

* pip install tensorflow==1.13.2 (tensorflow 설치 명령어)

필요한 모든 기본적인 패키지 설치는 완료되었습니다.

나중에 필요한 패키지가 생기면 activate된 상태로 설치하거나

Anaconda Prompt (설정된 가상환경 이름)로 접속하면 자동으로 activate된 상태가 되므로

이 상태에서 추가로 설치합니다.

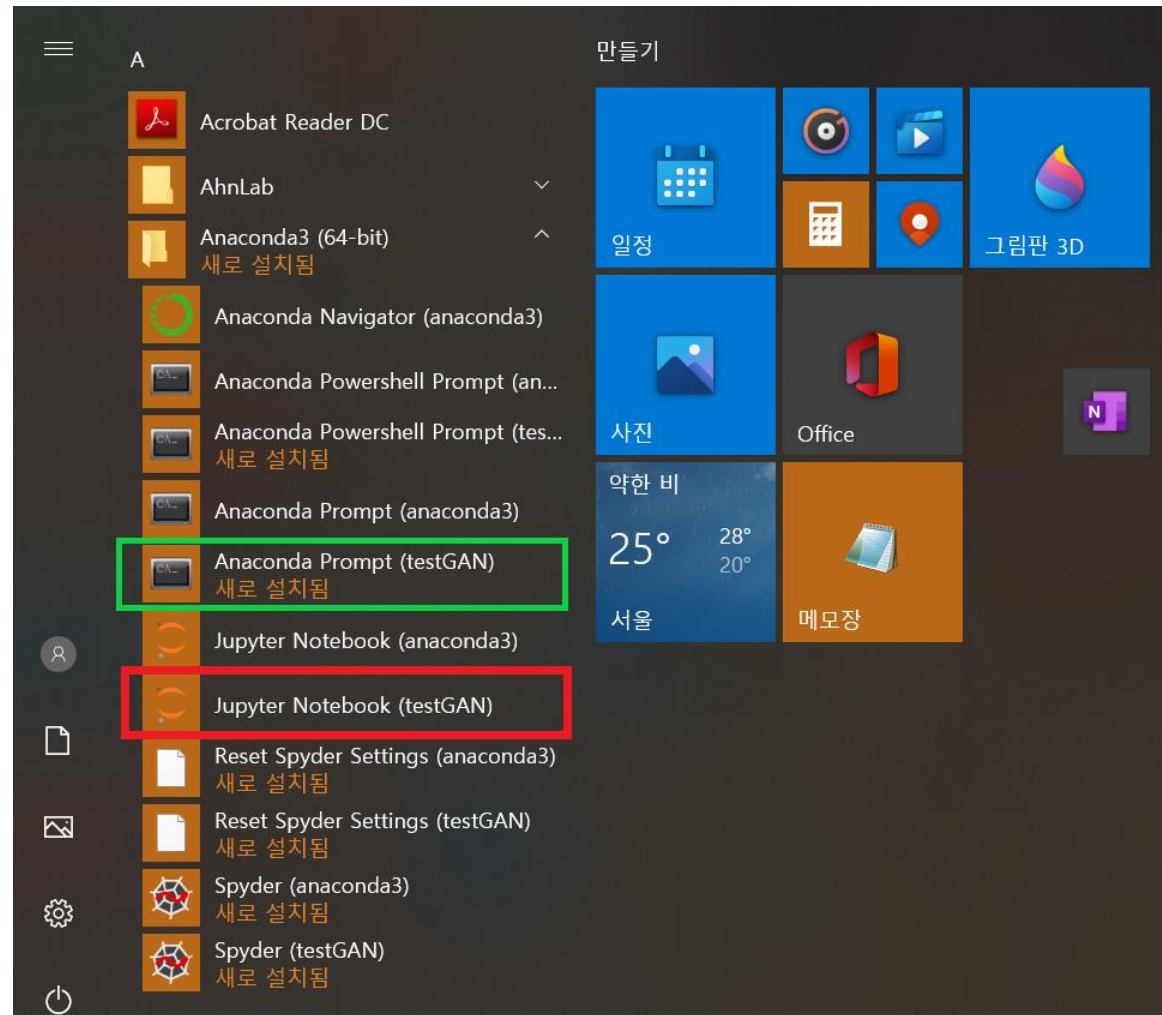
이제 파이썬으로 프로그래밍 하려면 파이썬 IDLE(Integrated DeveLopment Environment)가 필요한데,

여기서 우리는 Jupyter Notebook 이라는 IDLE를 사용합니다.

필요에 따라 해당 파일을 더 쉽게 실행하기 위해 바탕 화면에 바로 가기 아이콘을 생성합니다.



7. 앞으로 실행할 때는 Jupyter Notebook (testGAN)으로 표시된 IDE로 실행해야
변경된 버전과 패키지가 유지됩니다. (중요)



8. 마지막으로 Jupyter Notebook (testGAN)으로 표시된 IDE로 실행 후
keras나 tensorflow가 잘 설치되었는지 확인합니다.

```
In [1]: import keras

Using TensorFlow backend.
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:526: FutureWarning: Passing (type, 1) or '1type'
e' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype(("qint8", np.int8, 1))
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:527: FutureWarning: Passing (type, 1) or '1type'
e' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype(("qint8", np.uint8, 1))
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:528: FutureWarning: Passing (type, 1) or '1type'
e' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype(("qint16", np.int16, 1))
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:529: FutureWarning: Passing (type, 1) or '1type'
e' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype(("qint16", np.uint16, 1))
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:530: FutureWarning: Passing (type, 1) or '1type'
e' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype(("qint32", np.int32, 1))
C:\Users\User\Anaconda3\envs\testGAN\lib\site-packages\tensorflow\python\framework\dtypes.py:535: FutureWarning: Passing (type, 1) or '1type'
e' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype(("resource", np.ubyte, 1))

In [2]: print(keras.__version__)

2.2.4

In [3]: import tensorflow as tf

In [4]: print(tf.__version__)

1.13.2

In [ ]:
```

수고하셨습니다.