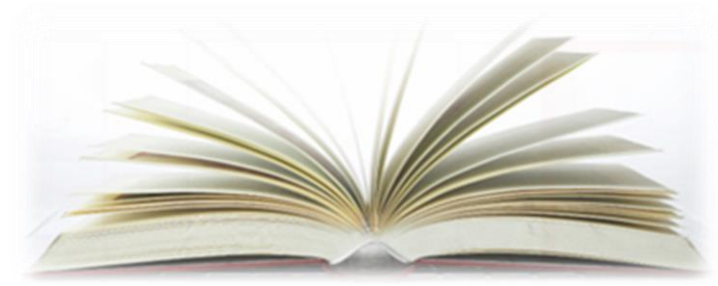


제조데이터 분석 및 시각화 ('22.02.22 - '22.02.25) -부록-

Prepared by DaeKyeong Kim
Ph.D.

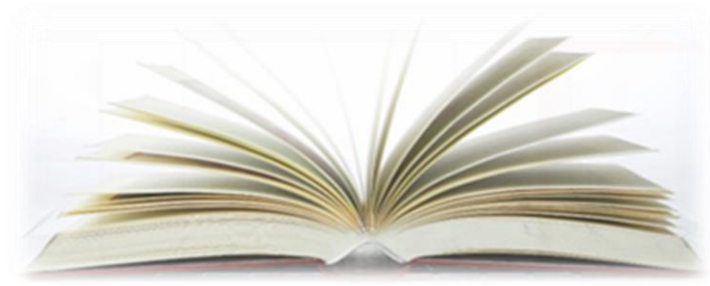


Contents



1.	제조데이터 분석 및 시각화를 위한 환경 구성	3
----	--------------------------------	---

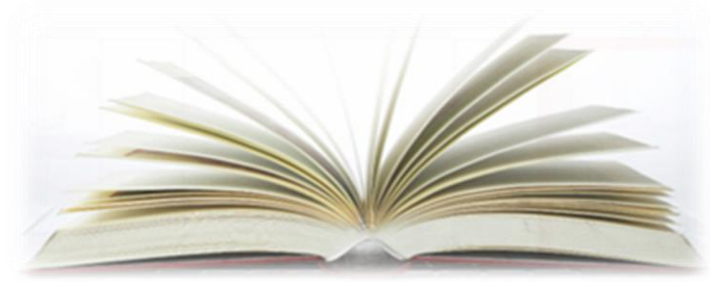
1차시



제조데이터 분석 및 시각화를 위한 환경 구성

1. 윈10+아나콘다+텐서플로2
2. Google Colab

학습목표



- ❖ 이 워크샵에서는 제조데이터 분석 및 시각화를 위한 환경 구성을 위한 내용을 다룬다.
- 윈10+아나콘다+텐서플로2를 설치하고 사용할 수 있다.
- 코랩에 접속하고 사용할 수 있다.

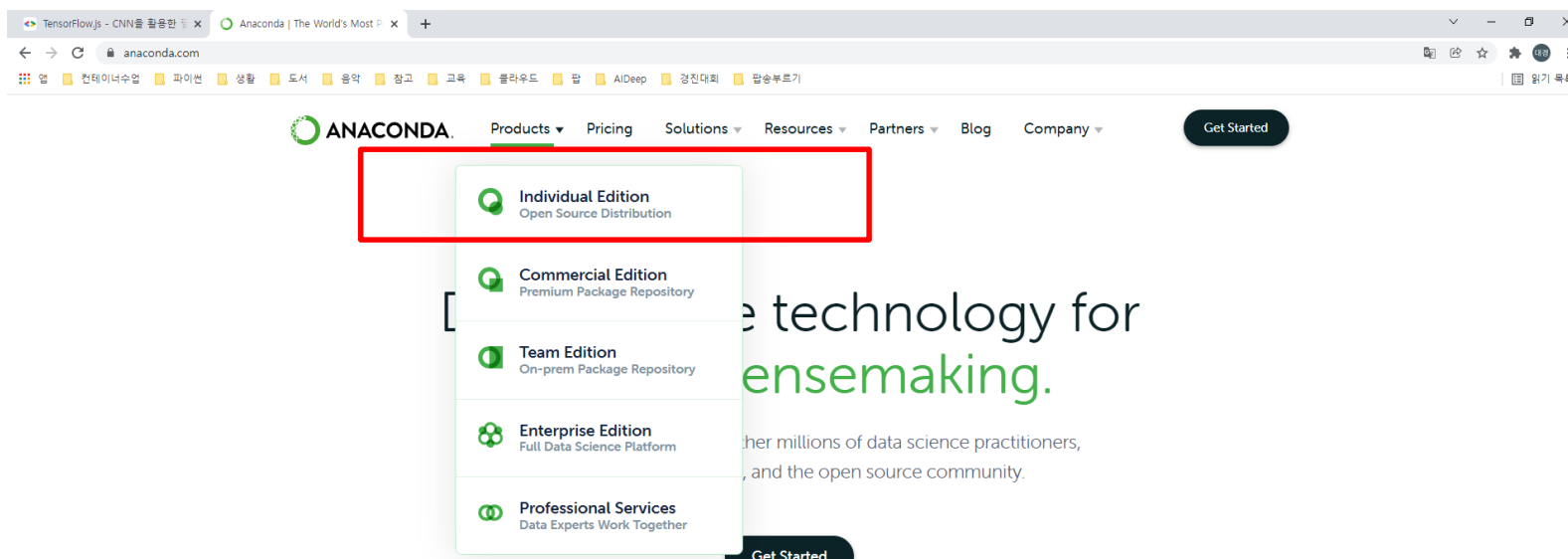
Unit 1



원10+아나콘다+텐서플로2

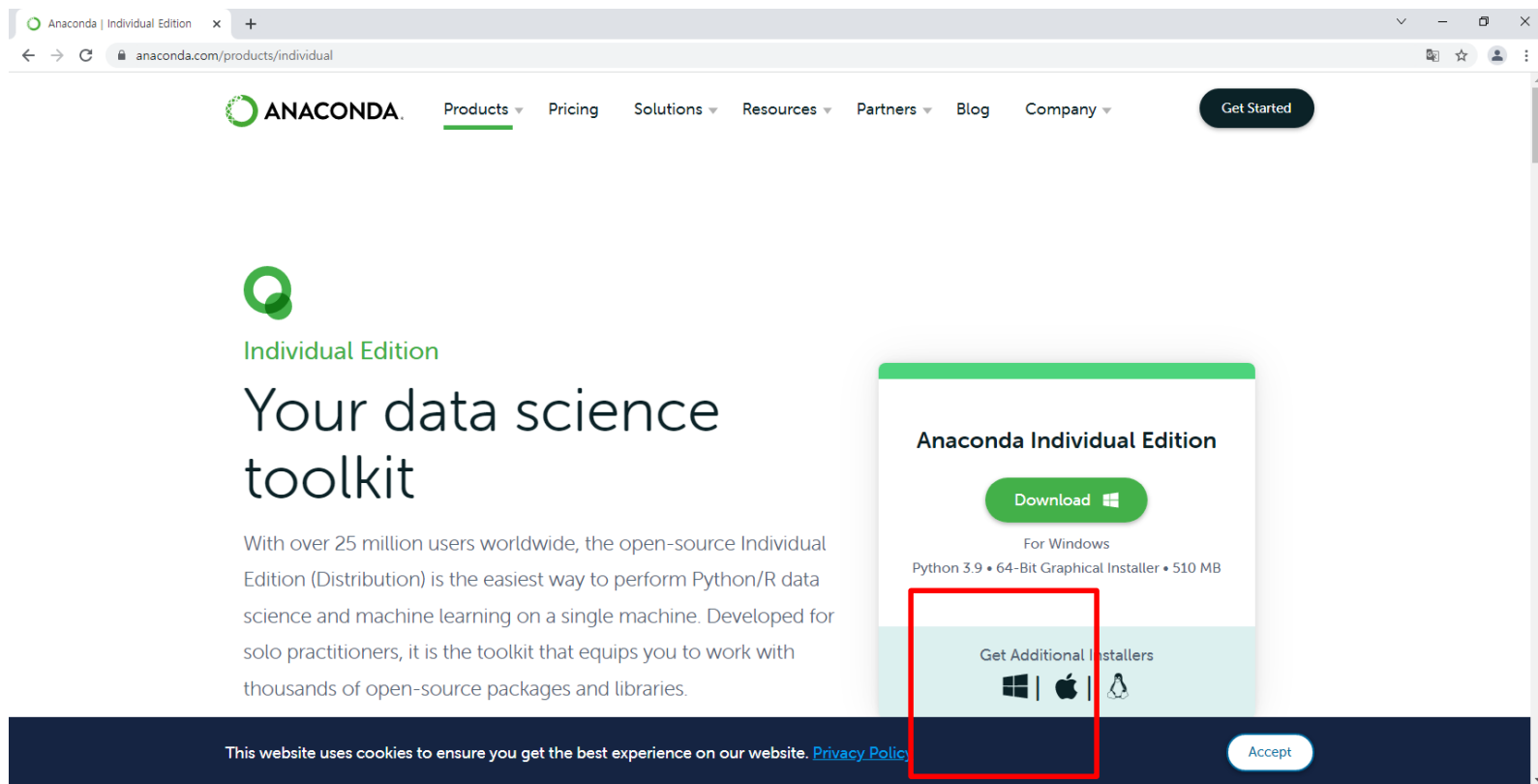
● <https://www.anaconda.com/>

❖ Individual Edition 선택



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

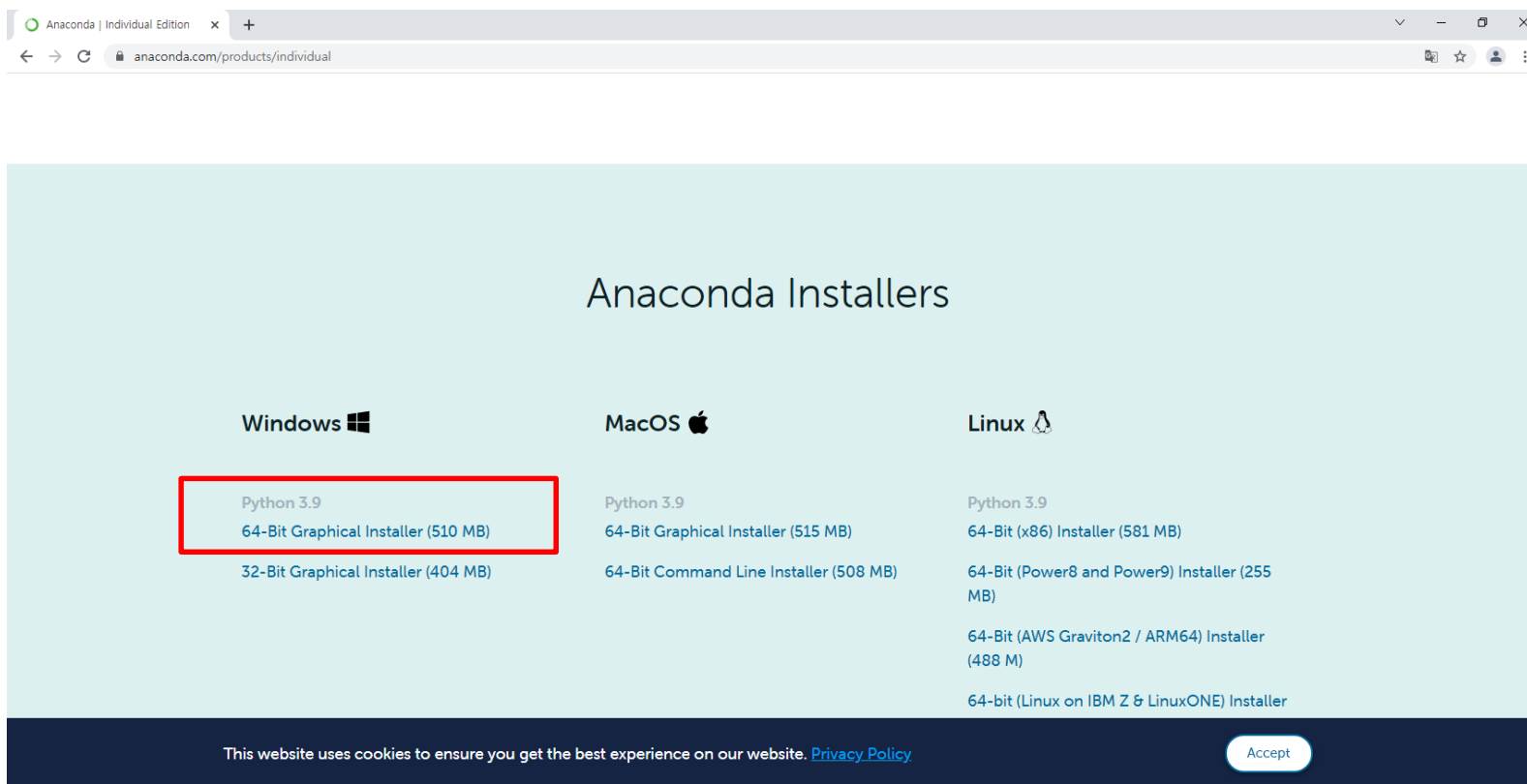
- ❖ Get Additional Installers > 윈도우 선택



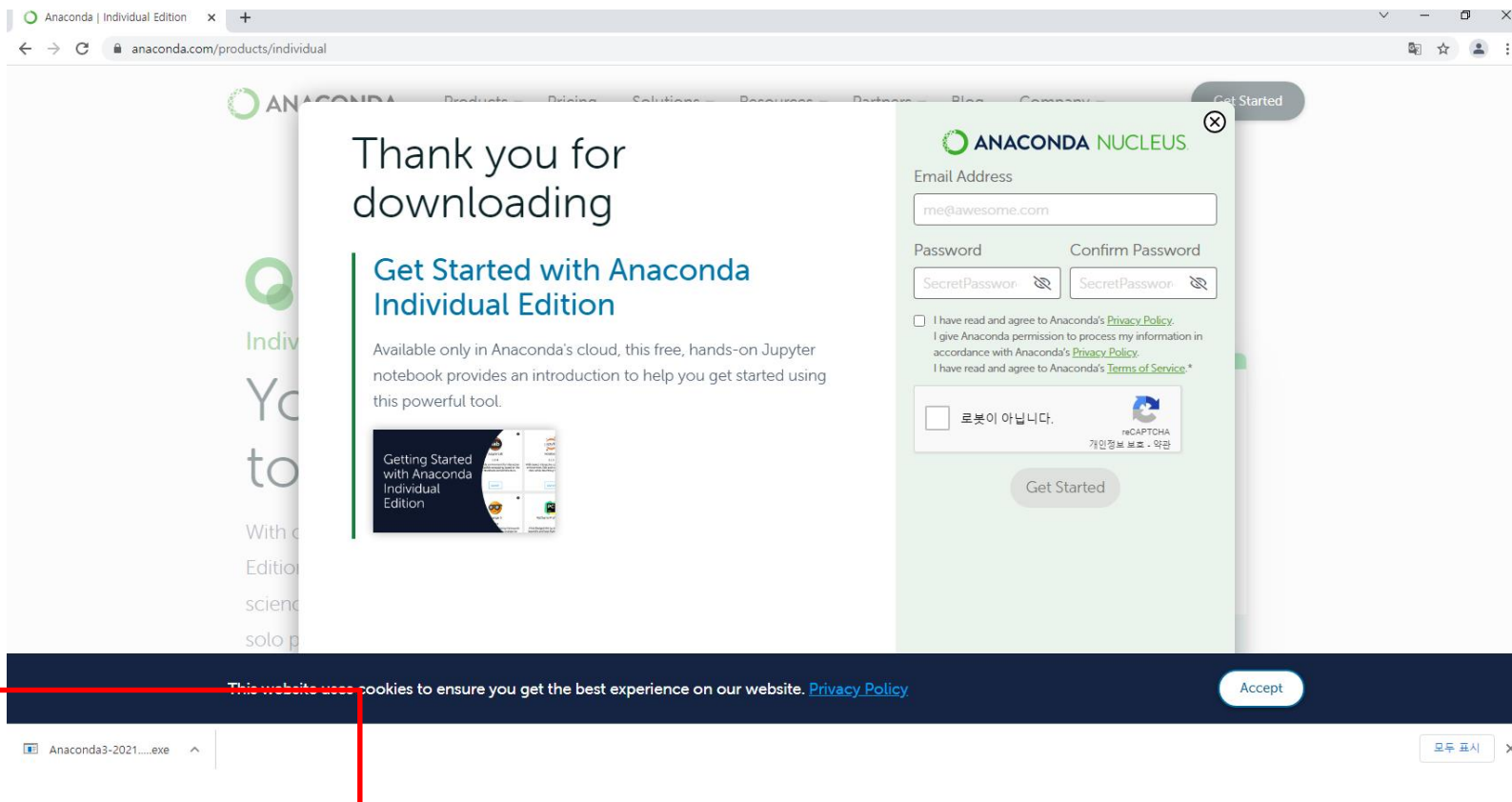
- <https://www.anaconda.com/>

- ❖ Python 3.9

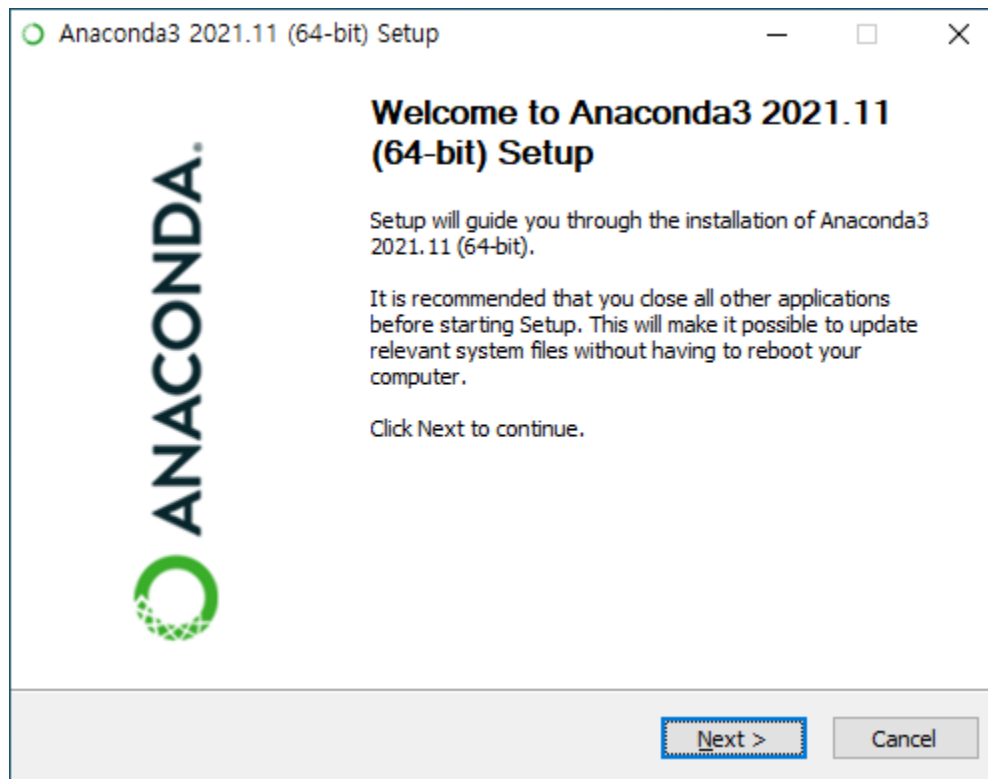
- 64-Bit Graphical Installer (510 MB) 클릭-2021년 12월 04일 기준



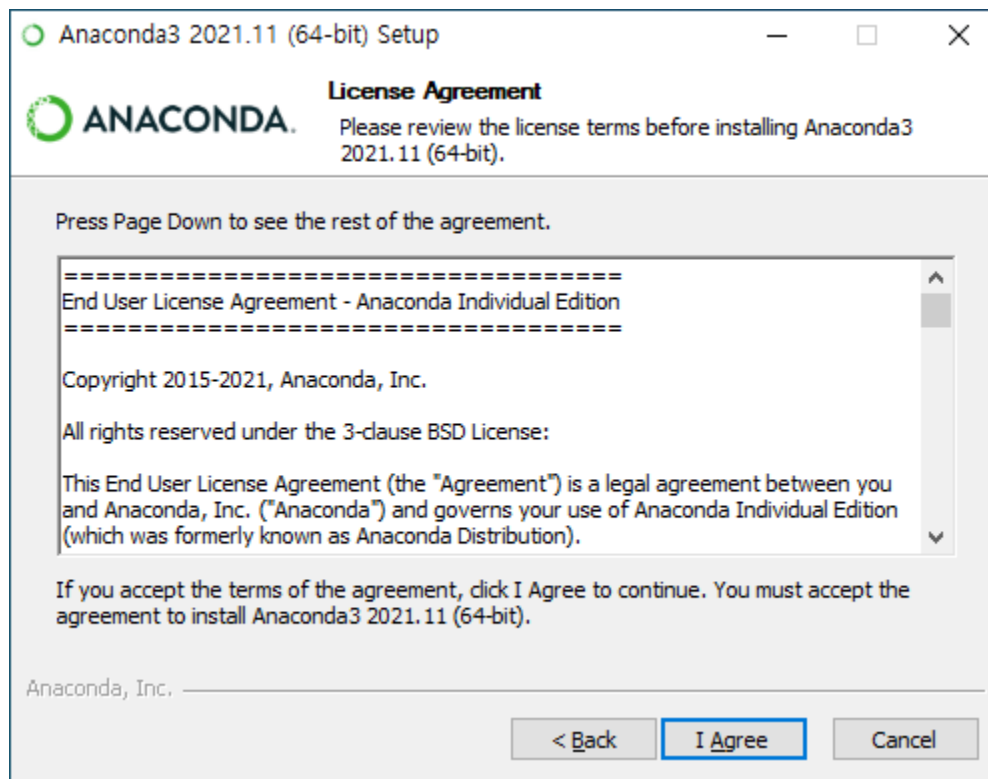
❖ 다운로드 중



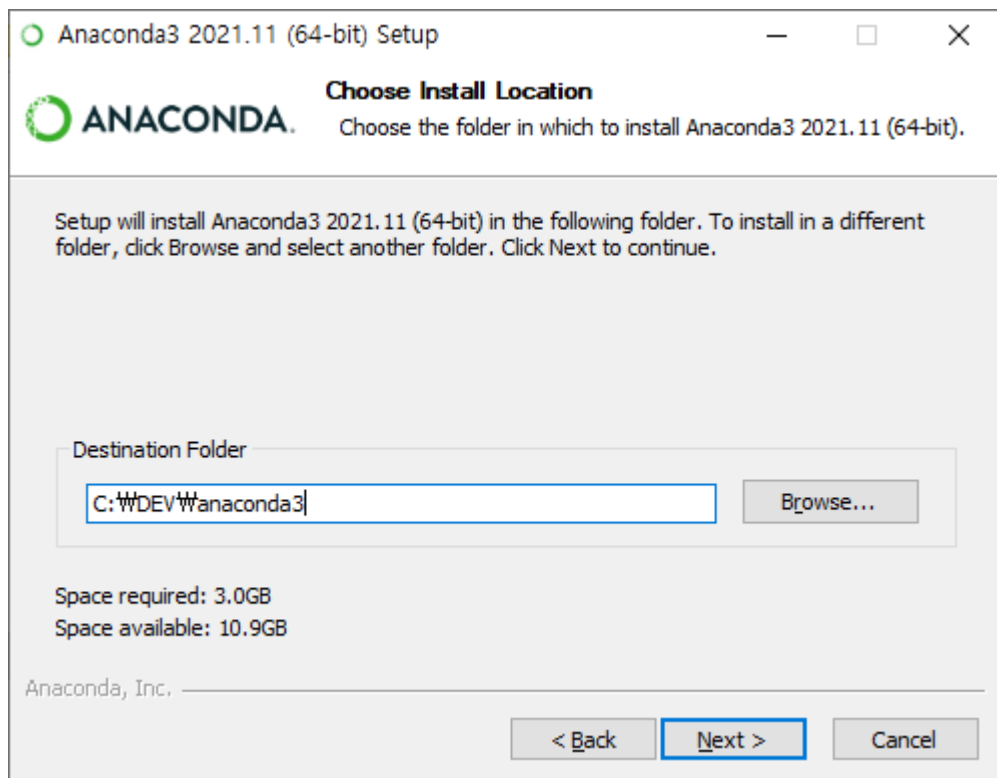
* Next 클릭



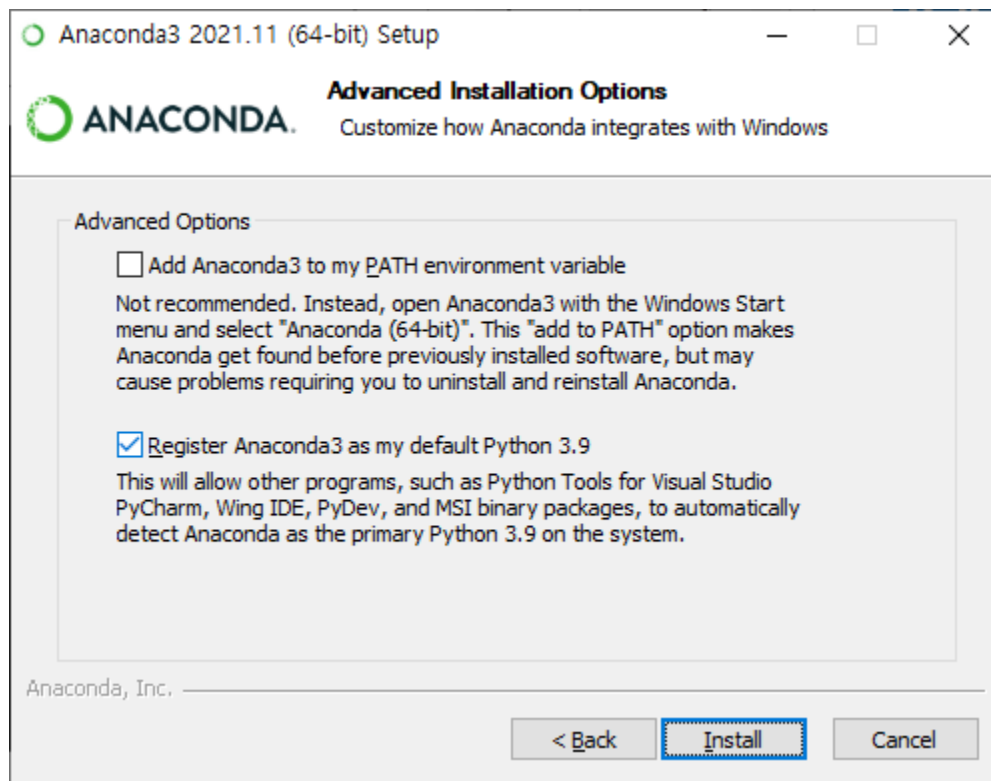
* I Agree 클릭



* C:\WDEV\Wanaconda3로 경로 설정하고 Next



- Register Anaconda3 as my default Python 3.8 ' 만 체크하고 Install 클릭



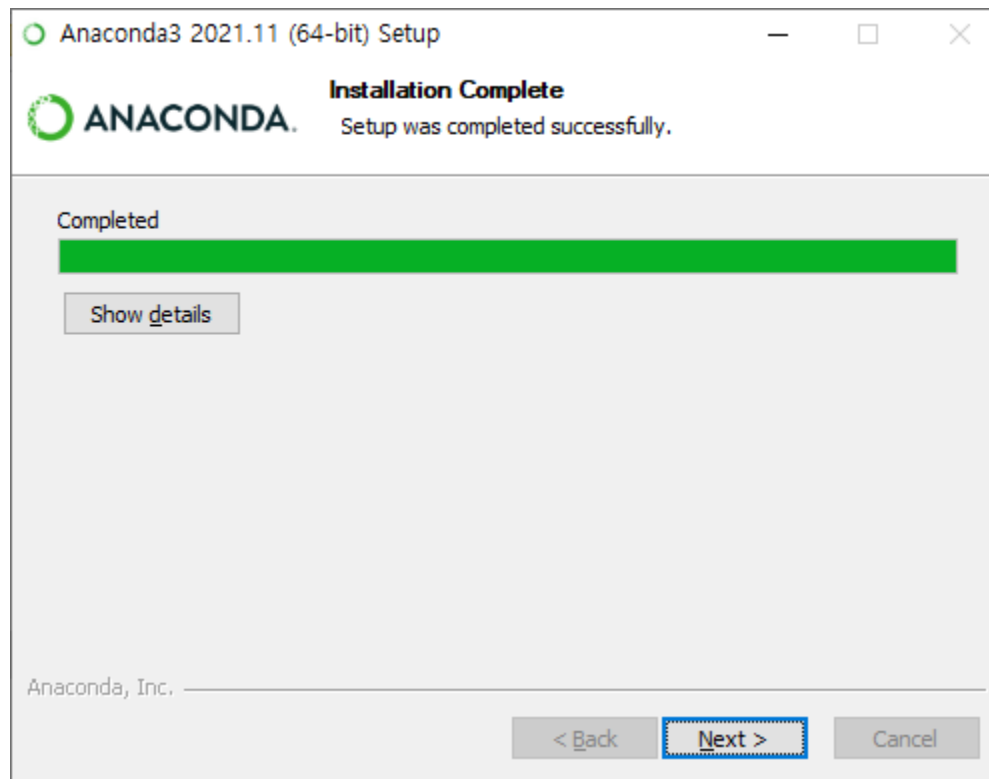
❖ Add Anaconda3 to my PATH environment variable:

- PATH 환경 변수에 아나콘다를 추가할지 선택
- 아나콘다 외에 다른 파이썬 인터프리터를 환경변수에 등록해서 사용한다면 체크 해제
- 아나콘다만을 사용하는 경우, 아나콘다가 주력일 경우로 윈도우 cmd 창에서 파이썬을 실행할 경우, 현재 PC에 파이썬을 설치한 적이 없는 경우에는 체크하지만, 아나콘다의 python.exe는 PATH에 추가하지 않더라도 시작 메뉴의 Anaconda Prompt를 통해 사용할 수 있다.
- (체크할 경우 윈도우 cmd 창 경로와 상관없이 아나콘다를 파이썬으로 인식)

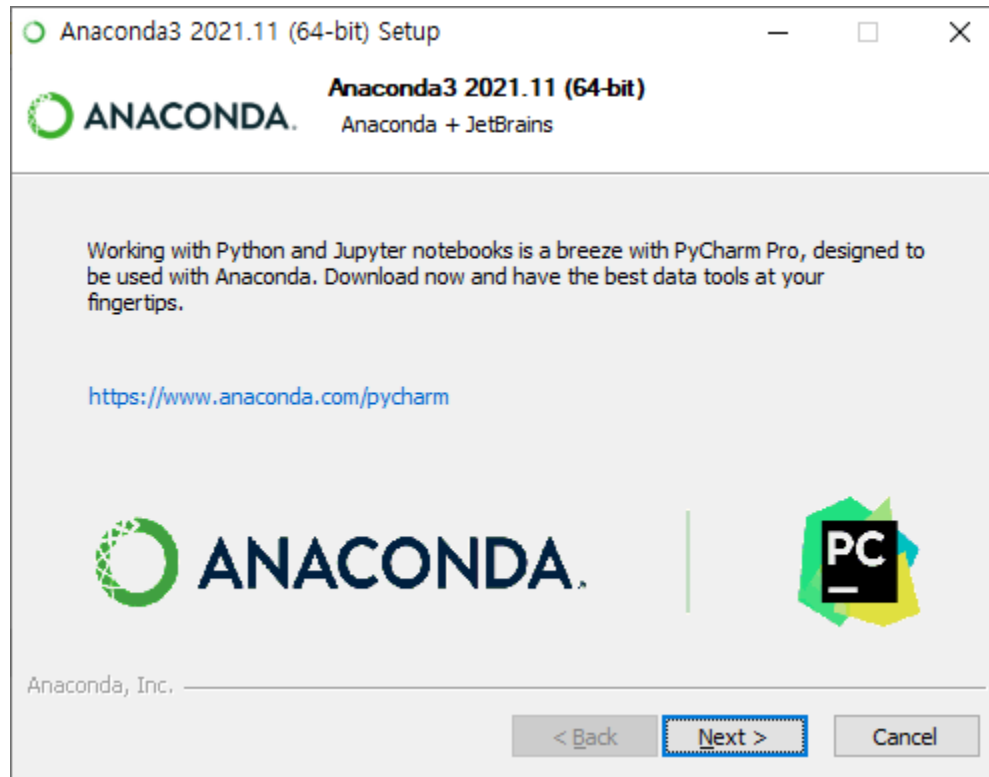
❖ Register Anaconda3 as my default Python 3.8:

- 아나콘다를 기본 파이썬으로 등록할지 여부를 선택
- (체크할 경우 개발 도구나 에디터에서 아나콘다를 파이썬으로 인식)

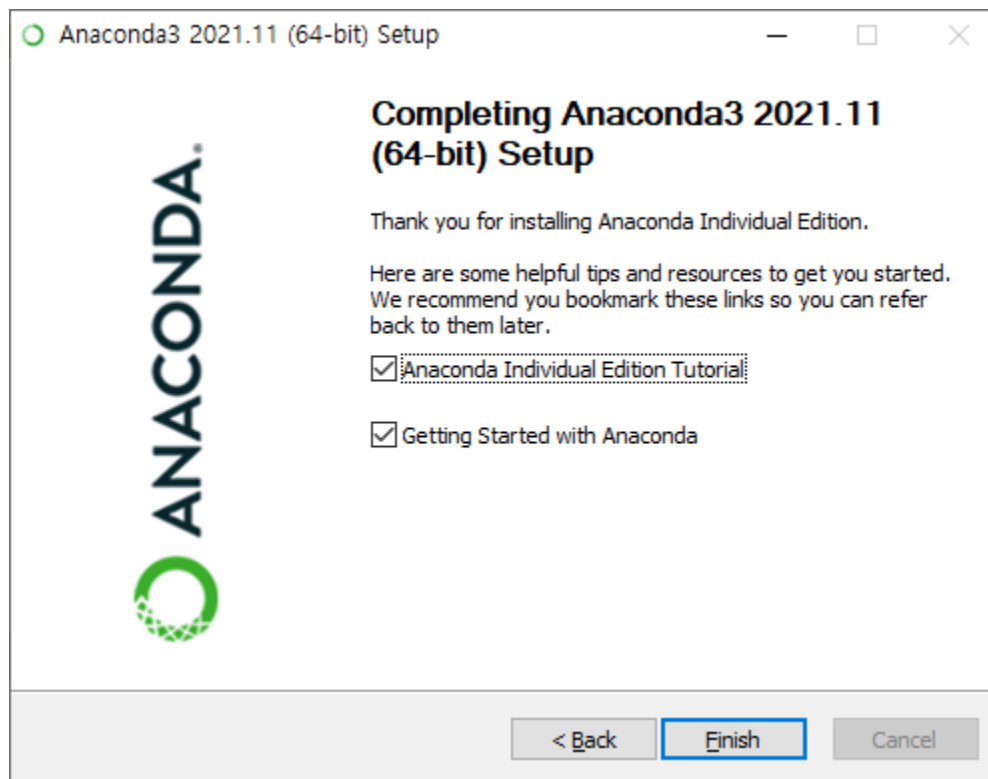
* Next 클릭



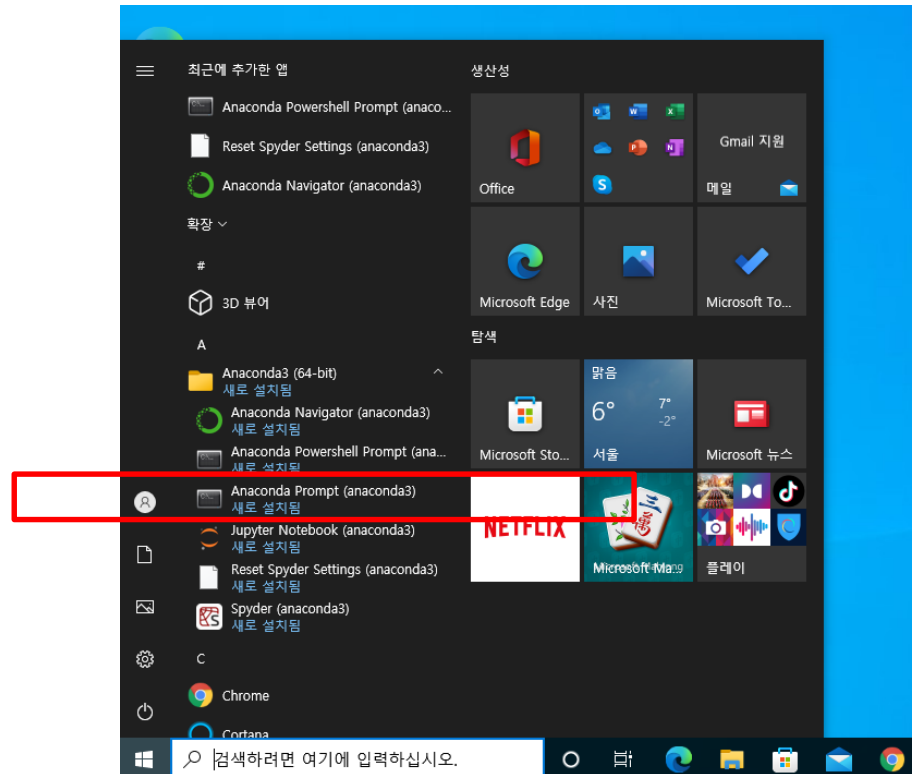
* Next 클릭



* Finish 클릭(설치 완료)



❖ * 시작 -> Anaconda3 (64-bit) -> Anaconda Prompt (anaconda3) 실행해서 간단하게 python 테스트해보기



❖ * 시작 -> Anaconda3 (64-bit) -> Anaconda Prompt (anaconda3) 실행해서 간단하게 python 테스트해보기

Anaconda Prompt (anaconda3)

```
(base) C:\Users\k8s>python -V  
Python 3.9.7
```

```
(base) C:\Users\k8s>_
```

- pip을 업그레이드

```
(base) C:\Users\Wapr0621>pip install --upgrade pip
```

❖ 설치 시 에러 메시지 발생할 수 있음. 무시하고 진행

- 가상환경 생성

```
(base) C:\Users\Wapr0621>conda create -n tf2_38 python=3.8
```

```
Collecting package metadata (current_repodata.json): done
```

```
Solving environment: done
```

```
## Package Plan ##
```

```
environment location: C:\DEV\anaconda3\envs\tensorflow
```

```
added / updated specs:
```

```
- pip  
- python=3.8
```

```
The following packages will be downloaded:
```

package	build	
certifi-2021.10.8	py38haa95532_0	152 KB
pip-21.2.2	py38haa95532_0	1.9 MB
python-3.8.12	h6244533_0	16.0 MB
setuptools-58.0.4	py38haa95532_0	779 KB
wincertstore-0.2	py38haa95532_2	15 KB
Total:		18.8 MB

The following NEW packages will be INSTALLED:

```
ca-certificates  pkgs/main/win-64::ca-certificates-2021.10.26-haa95532_2
certifi          pkgs/main/win-64::certifi-2021.10.8-py38haa95532_0
openssl          pkgs/main/win-64::openssl-1.1.1l-h2bbff1b_0
pip              pkgs/main/win-64::pip-21.2.2-py38haa95532_0
python           pkgs/main/win-64::python-3.8.12-h6244533_0
setuptools       pkgs/main/win-64::setuptools-58.0.4-py38haa95532_0
sqlite           pkgs/main/win-64::sqlite-3.36.0-h2bbff1b_0
vc               pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime   pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel            pkgs/main/noarch::wheel-0.37.0-pyhd3eb1b0_1
wincertstore     pkgs/main/win-64::wincertstore-0.2-py38haa95532_2
```

Proceed ([y]/n)?

...

done

#

To activate this environment, use

#

\$ conda activate tensorflow

#

To deactivate an active environment, use

#

\$ conda deactivate

(base) C:\Users\Wapr0621>

- 가상환경으로 이동

```
(base) C:\Users\Wapr0621>conda activate tf2_38
```

```
(tf2_38) C:\Users\Wapr0621>
```

- CPU 버전인 경우

```
(tf2_38) C:\Users\Wapr0621>
```


- 텐서플로우 설치 테스트

(tf2_38) C:\WUsers\Wapr0621>python

Python 3.8.12 (default, Oct 12 2021, 03:01:40) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import tensorflow as tf

>>> with tf.compat.v1.Session() as sess:

... helloworld = tf.constant("Hello World!")

... print(sess.run(helloworld))

...

2021-12-04 18:35:34.557406: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

b'Hello World!'

>>> helloworld = tf.constant("Hello World!")

>>> tf.print(helloworld)

Hello World!

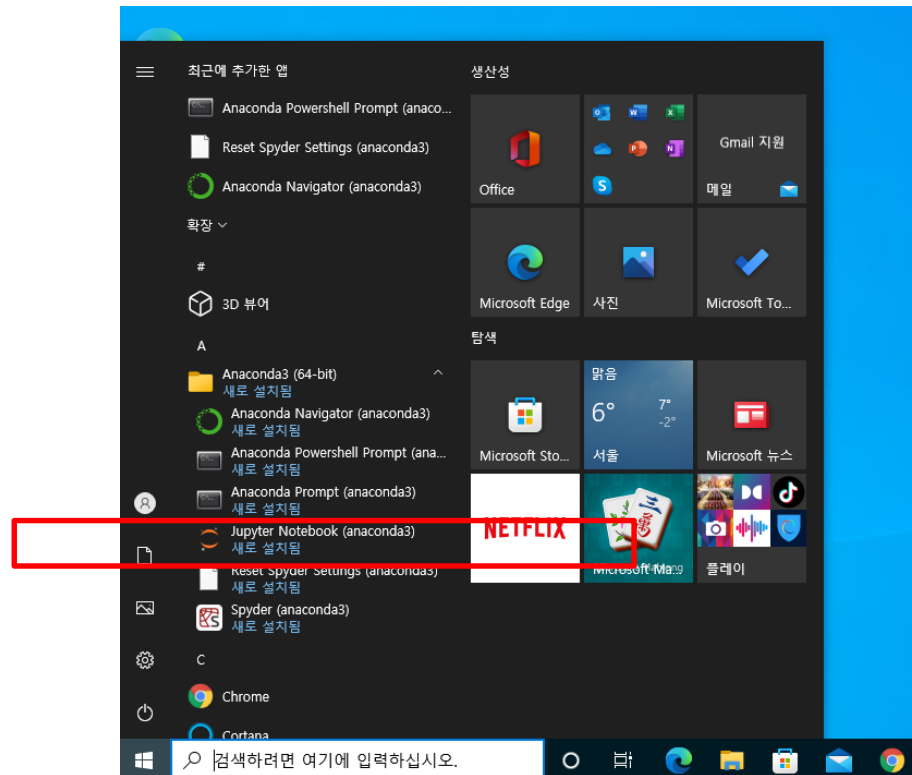
>>> quit()

(tf2_38) C:\WUsers\Wapr0621>

2칸 띄어쓰기

엔터

❖ * 시작 -> Anaconda3 (64-bit) -> Jupyter Notebook(anaconda3) 실행



❖ * 시작 -> Anaconda3 (64-bit) -> Jupyter Notebook(anaconda3) 실행

The image shows a Jupyter Notebook terminal window titled "Jupyter Notebook (anaconda3)" with the following output:

```
[W 2021-12-04 18:40:17.415 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 2021-12-04 18:40:17.415 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 2021-12-04 18:40:17.431 LabApp] JupyterLab extension loaded from C:\DEV\anaconda3\lib\site-packages\jupyterlab
[I 2021-12-04 18:40:17.431 LabApp] JupyterLab application directory is C:\DEV\anaconda3\share\jupyter\lab
[I 18:40:17.431 NotebookApp] Serving notebooks from local directory: C:\Users\k8s
[I 18:40:17.431 NotebookApp] Jupyter Notebook 6.4.5 is running at:
[I 18:40:17.431 NotebookApp] http://localhost:8888/?token=cfa292e686aeba5f2980e1f907db62ad38
[I 18:40:17.431 NotebookApp] or http://127.0.0.1:8888/?token=cfa292e686aeba5f2980e1f907db62ad38
[I 18:40:17.431 NotebookApp] Use Control-C to stop this server and shut down all open notebooks.
[C 18:40:17.477 NotebookApp]
```

Below the terminal output, the text reads:

```
To access the notebook, open this file in a browser:
file:///C:/Users/k8s/AppData/Roaming/jupyter/runtime/nbserver-4...
Or copy and paste one of these URLs:
http://localhost:8888/?token=cfa292e686aeba5f2980e1f907db62ad38
or http://127.0.0.1:8888/?token=cfa292e686aeba5f2980e1f907db62ad38
```

Overlaid on the right side of the terminal is a Windows file opening dialog box. The text in the dialog box is:

이 파일을 열 때 사용할 앱을 선택하세요.

계속 이 앱 사용

Microsoft Edge
Microsoft의 새 브라우저에서 더 많은 온라인 기능을
사용하세요.

기타 옵션

Chrome
새로 만들기

Internet Explorer

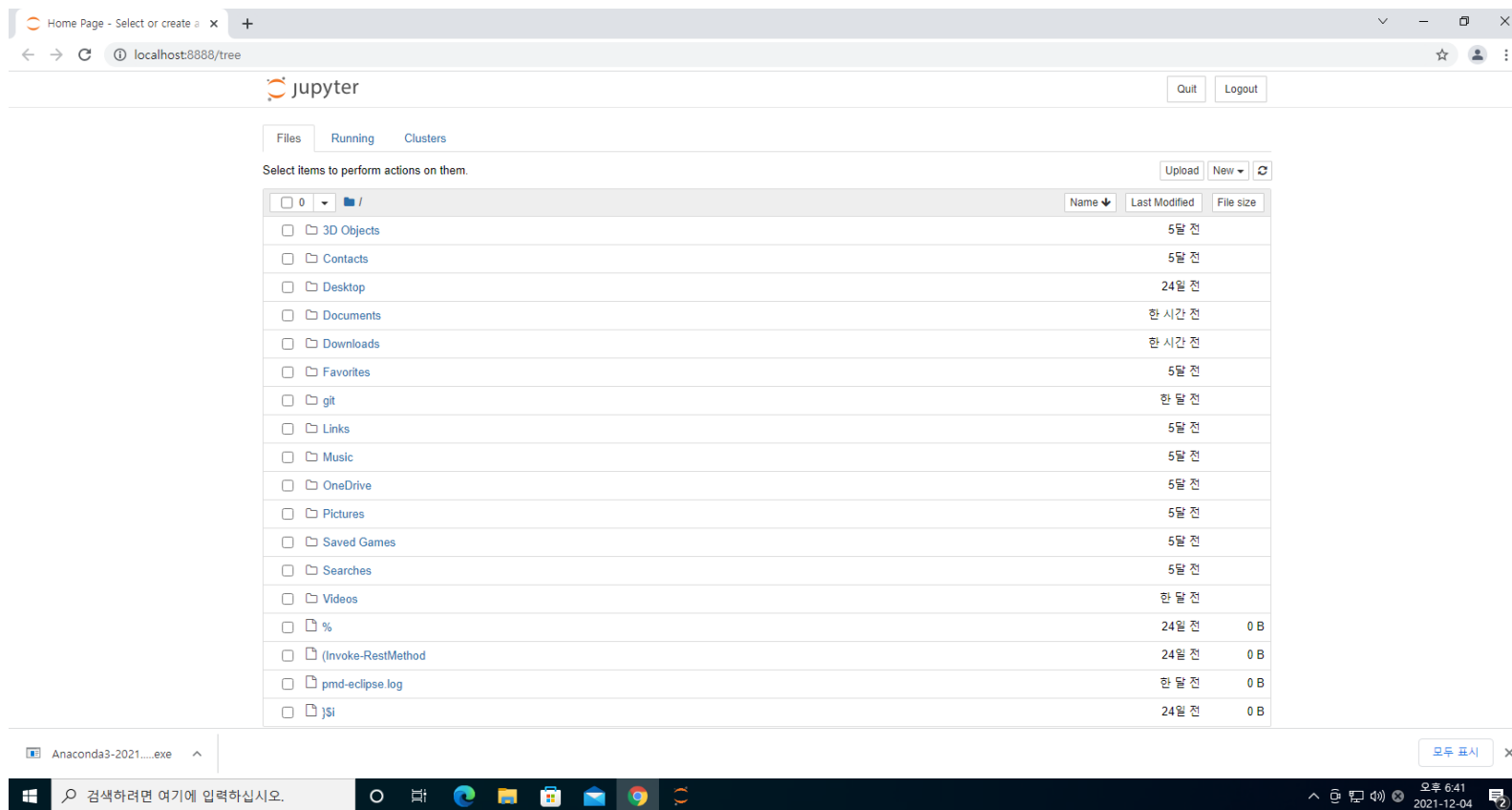
Microsoft Store에서 앱 찾기

추가 앱 ↓

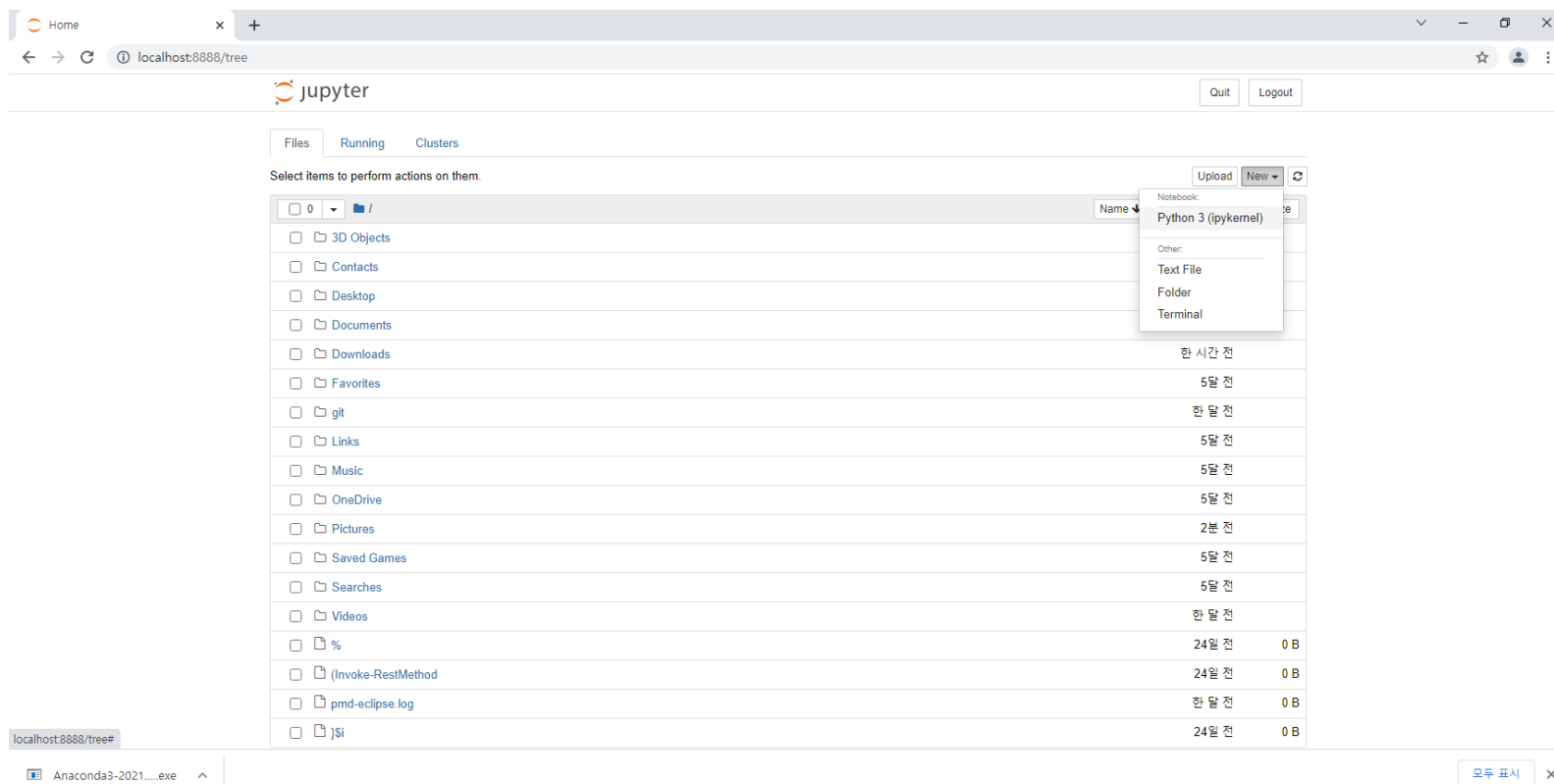
☐ 항상 이 앱을 사용하여 .html 파일 열기

확인

❖ * 시작 -> Anaconda3 (64-bit) -> Jupyter Notebook(anaconda3) 실행



❖ 메인화면에서 오른쪽 상단부분에 New 를 눌러 Python3 로 새파일 생성



그런데 !!!

주피터 노트북에는 파이
썬 커널이 [default]만 보
임

❖ Jupyter notebook 설치

(tf2_38) C:\Users\Wapr0621>pip install jupyter

Collecting jupyter

Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)

Collecting nbconvert

Downloading nbconvert-6.3.0-py3-none-any.whl (556 kB)

...

Installing collected packages: pyparsing, pycparser, jsonschema, ipython-genutils, webencodings, packaging, nbformat, MarkupSafe, cffi, testpath, pywinpty, pandocfilters, nbclient, mistune, jupyterlab-pygments, Jinja2, defusedxml, bleach, argon2-cffi-bindings, terminado, Send2Trash, prometheus-client, nbconvert, argon2-cffi, notebook, widgetsnbextension, qtpy, jupyterlab-widgets, qtconsole, jupyter-console, ipywidgets, jupyter

Successfully installed MarkupSafe-2.0.1 Send2Trash-1.8.0 argon2-cffi-21.2.0 argon2-cffi-bindings-21.2.0 attrs-21.2.0 bleach-4.1.0 cffi-1.15.0 defusedxml-0.7.1 importlib-resources-5.4.0 ipython-genutils-0.2.0 ipywidgets-7.6.5 Jinja2-3.0.3 jsonschema-4.2.1 jupyter-1.0.0 jupyter-console-6.4.0 jupyterlab-pygments-0.1.2 jupyterlab-widgets-1.0.2 mistune-0.8.4 nbclient-0.5.9 nbconvert-6.3.0 nbformat-5.1.3 notebook-6.4.6 packaging-21.3 pandocfilters-1.5.0 prometheus-client-0.12.0 pycparser-2.21 pyparsing-3.0.6

pyparsing-3.0.6 pywinpty-1.1.6 qtconsole-5.2.1 qtpy-1.11.3 terminado-0.12.1 testpath-0.5.0 webencodings-0.5.1 widgetsnbextension-3.5.2 **(tf2_38)**

C:\Users\Wapr0621>

❖ Jupyter notebook에 가상환경 커널 추가하기

(tf2_38) C:\WUsers\Wapr0621>pip install ipykernel

Collecting ipykernel

Downloading ipykernel-6.6.0-py3-none-any.whl (126 kB)

 126 kB 6.4 MB/s

Collecting tornado<7.0,>=4.2

Downloading tornado-6.1-cp38-cp38-win_amd64.whl (422 kB)

...

Collecting wcwidth

Downloading wcwidth-0.2.5-py2.py3-none-any.whl (30 kB)

Requirement already satisfied: six>=1.5 in c:\wdev\anaconda3\envs\tf2_38\lib\site-packages (from python-dateutil>=2.1->jupyter-client<8.0->ipykernel) (1.16.0)

Installing collected packages: wcwidth, traitlets, pywin32, parso, tornado, pyzmq, python-dateutil, pygments, prompt-toolkit, pickleshare, nest-asyncio, matplotlib-inline, jupyter-core, jedi, entrypoints, decorator, colorama, backcall, jupyter-client, ipython, debugpy, ipykernel

Successfully installed backcall-0.2.0 colorama-0.4.4 debugpy-1.5.1 decorator-5.1.0 entrypoints-0.3 ipykernel-6.6.0 ipython-7.30.1 jedi-0.18.1 jupyter-client-7.1.0 jupyter-core-4.9.1 matplotlib-inline-0.1.3 nest-asyncio-1.5.4 parso-0.8.3 pickleshare-0.7.5 prompt-toolkit-3.0.24 pygments-2.10.0 python-dateutil-2.8.2 pywin32-302 pyzmq-22.3.0 tornado-6.1 traitlets-5.1.1 wcwidth-0.2.5

(tf2_38) C:\WUsers\Wapr0621>

Jupyter Notebook

(tf2_38) C:\Users\Wapr0621>python -m ipykernel install --user --name tf2_38 --display-name "tf2_38"

Installed kernelspec tf2_38 in C:\Users\Wapr0621\AppData\Roaming\jupyter\kernels\tf2_38

(tf2_38) C:\Users\Wapr0621>jupyter notebook

```
[I 2021-12-04 18:56:05.621 LabApp] JupyterLab extension loaded from C:\DEV\Wanaconda3\lib\site-packages\jupyterlab
[I 2021-12-04 18:56:05.621 LabApp] JupyterLab application directory is C:\DEV\Wanaconda3\share\jupyter\lab
[I 18:56:05.621 NotebookApp] Serving notebooks from local directory: C:\Users\Wapr0621>
[I 18:56:05.621 NotebookApp] Jupyter Notebook 6.4.5 is running at:
[I 18:56:05.621 NotebookApp] http://localhost:8888/?token=f3e22a560ac5177c02a9fd0ea19df9fde5d28b3daed8bf05
[I 18:56:05.621 NotebookApp] or http://127.0.0.1:8888/?token=f3e22a560ac5177c02a9fd0ea19df9fde5d28b3daed8bf05
[I 18:56:05.621 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:56:05.684 NotebookApp]
```

To access the notebook, open this file in a browser:

`file:///C:/Users/k8s/AppData/Roaming/jupyter/runtime/nbserver-7100-open.html`

Or copy and paste one of these URLs:

`http://localhost:8888/?token=f3e22a560ac5177c02a9fd0ea19df9fde5d28b3daed8bf05`

or `http://127.0.0.1:8888/?token=f3e22a560ac5177c02a9fd0ea19df9fde5d28b3daed8bf05`

❖ 가상환경 커널 tf2_38이 잘 보임

The screenshot shows the Jupyter Notebook interface in a web browser. The browser's address bar displays 'localhost:8888/tree'. The Jupyter logo is visible at the top left of the interface, with 'Quit' and 'Logout' buttons to its right. Below the logo, there are tabs for 'Files', 'Running', and 'Clusters'. A message 'Select items to perform actions on them.' is displayed above a file browser table. The file browser table lists various folders and files, including '3D Objects', 'Contacts', 'Documents', 'Downloads', 'Favorites', 'Links', 'Music', 'nodeserver', 'OneDrive', 'OneDrive - (주)엘케이랩', 'Saved Games', 'Searches', and 'Videos'. A context menu is open over the file browser, showing options like 'Python 3 (ipykernel)' and 'tf2_38', which are highlighted with a red box. Other options in the menu include 'Text File', 'Folder', and 'Terminal'.

Name	Size	Modified
Python 3 (ipykernel)	38	8달 전
tf2_38		8달 전
Other:		
Text File		4달 전
Folder		4달 전
Terminal		2시간 전
		8달 전
		8달 전
		7달 전

❖ 가상환경 커널 tf38에서 텐서플로 코딩

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

2.8.0

❖ 소스 코드

```
import tensorflow as tf
```

```
print(tf.__version__)
```

❖ 가상환경 커널 tf38에서 텐서플로 코딩

```
In [1]: import tensorflow as tf
```

```
In [2]: with tf.compat.v1.Session() as sess:  
        helloworld = tf.constant("Hello World!")  
        print(sess.run(helloworld))
```

```
b'Hello World!'
```

```
In [3]: helloworld = tf.constant("Hello World!")  
        tf.print(helloworld)
```

```
Hello World!
```

❖ 소스 코드

```
import tensorflow as tf
```

```
with tf.compat.v1.Session() as sess:
```

```
    helloworld = tf.constant("Hello World!")
```

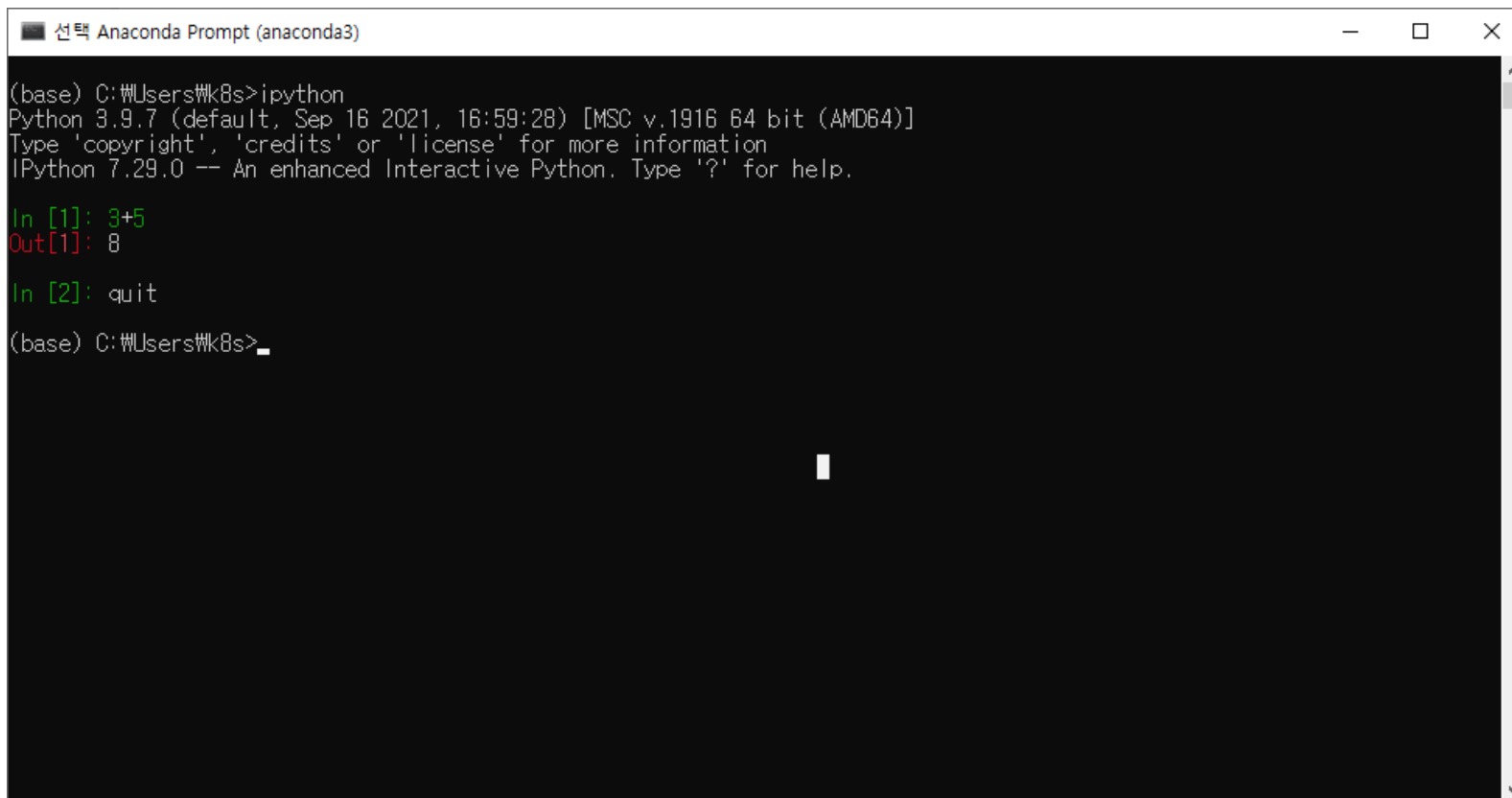
```
    print(sess.run(helloworld))
```

```
+++++
```

```
helloworld = tf.constant("Hello World!")
```

```
tf.print(helloworld)
```

❖ 아이파이썬 동작 확인



```
선택 Anaconda Prompt (anaconda3)
(base) C:\Users\kk8s>ipython
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.29.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: 3+5
Out[1]: 8

In [2]: quit
(base) C:\Users\kk8s>_
```

Collecting pandas

10.6 MB ...

Collecting matplotlib

7.2 MB 3.3 MB/s

Collecting scipy

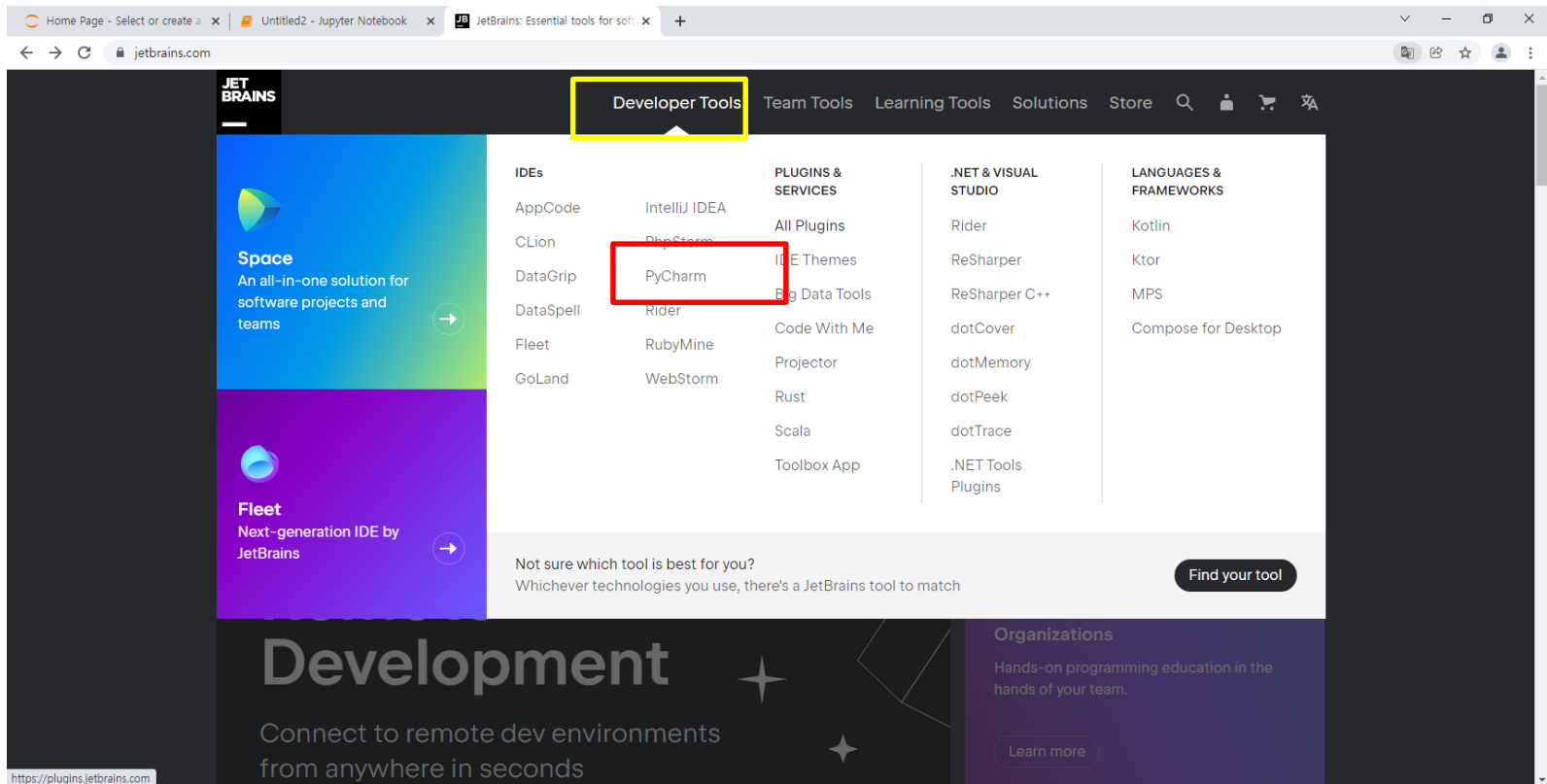
 36.9 MB 114 kB/s

...

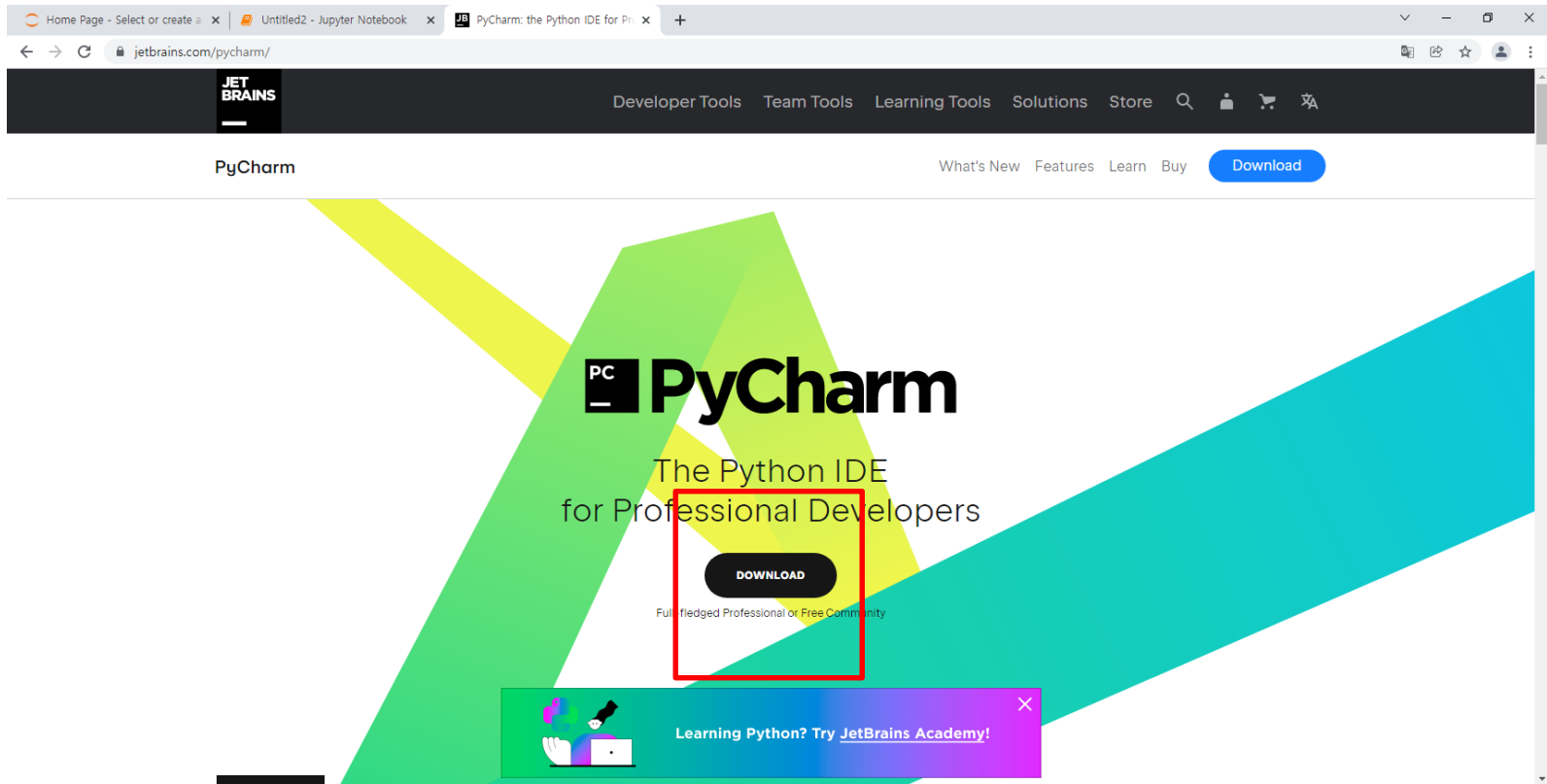
Successfully installed cyciler-0.11.0 fonttools-4.29.1 imageio-2.16.0 joblib-1.1.0 kiwisolver-1.3.2 matplotlib-3.5.1 mglearn-0.1.9 pandas-1.4.1 pillow-9.0.1 pytz-2021.3 scikit-learn-1.0.2 scipy-1.8.0 sklearn-0.0 threadpoolctl-3.1.0

```
(tf2_38) C:\DEV\pythonProject>
```

- ❖ PyCharm 홈페이지를 통해 커뮤니티 버전
- ❖ <https://www.jetbrains.com/>에서 Developer Tools메뉴 클릭



❖ PyCharm 홈페이지를 통해 커뮤니티 버전을 받아준다.



❖ PyCharm 홈페이지를 통해 커뮤니티 버전을 받아준다.

JetBrains s.r.o. [CZ] | <https://www.jetbrains.com/pycharm/download/#section=windows>

Tools Languages Solutions Support Store

PyCharm What's New Features Docs & Demos Buy Download

PC

Version: 2018.3.2
Build: 183.4886.43
Released: December 19, 2018

[System requirements](#)
[Installation instructions](#)
[Previous versions](#)

Download PyCharm

Windows macOS Linux

Professional

Full-featured IDE for Python & Web development

DOWNLOAD

Free trial

Community

Lightweight IDE for Python & Scientific development

DOWNLOAD

Free, open-source

Get the **Toolbox App** to download PyCharm and its future updates with ease

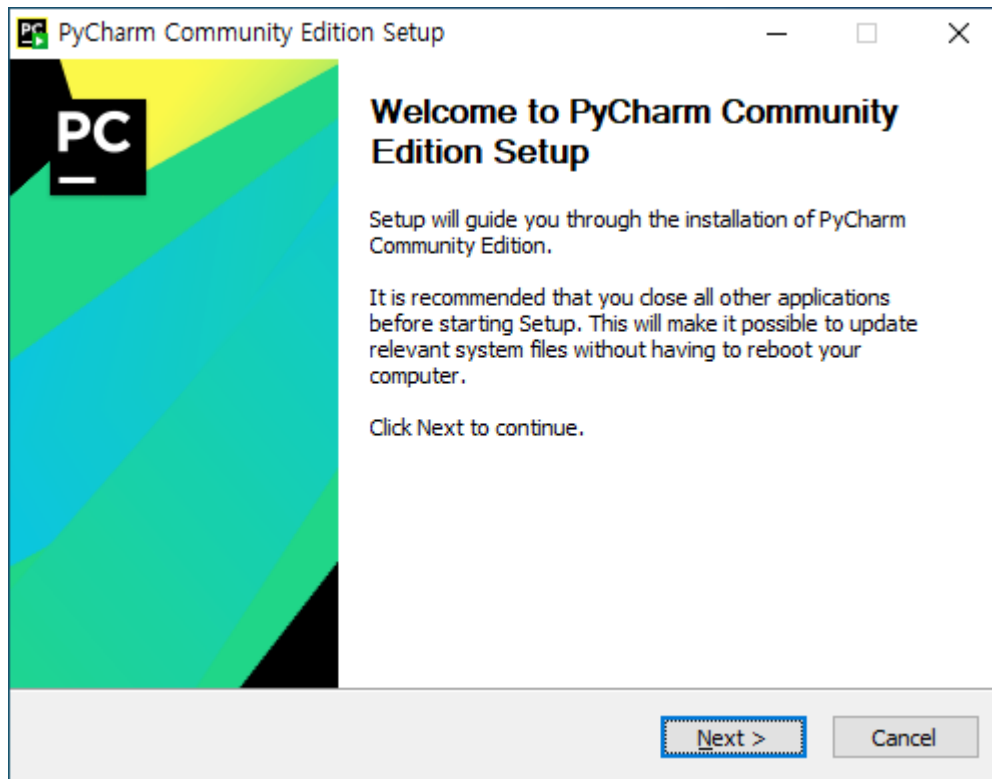
Cookies and IP addresses allow us to deliver and improve web content and to provide you with a personalized experience. Our website uses cookies and collects your IP address for these purposes.

JetBrains may use cookies and my IP address to collect individual statistics and to provide me with personalized offers and ads subject to the [Privacy Policy](#) and the [Terms of Use](#). JetBrains may use third-party services for this purpose. I can revoke my consent at any time by visiting the [Out-Opt page](#).

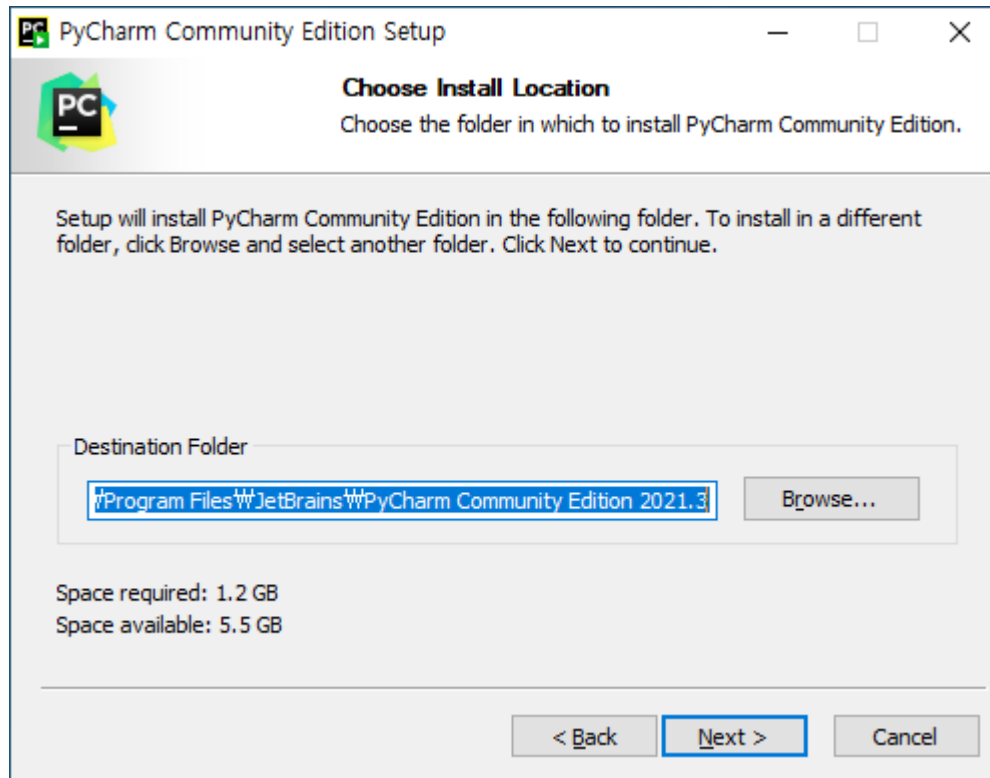
[Y]es, I agree [N]o, thanks

~ root#

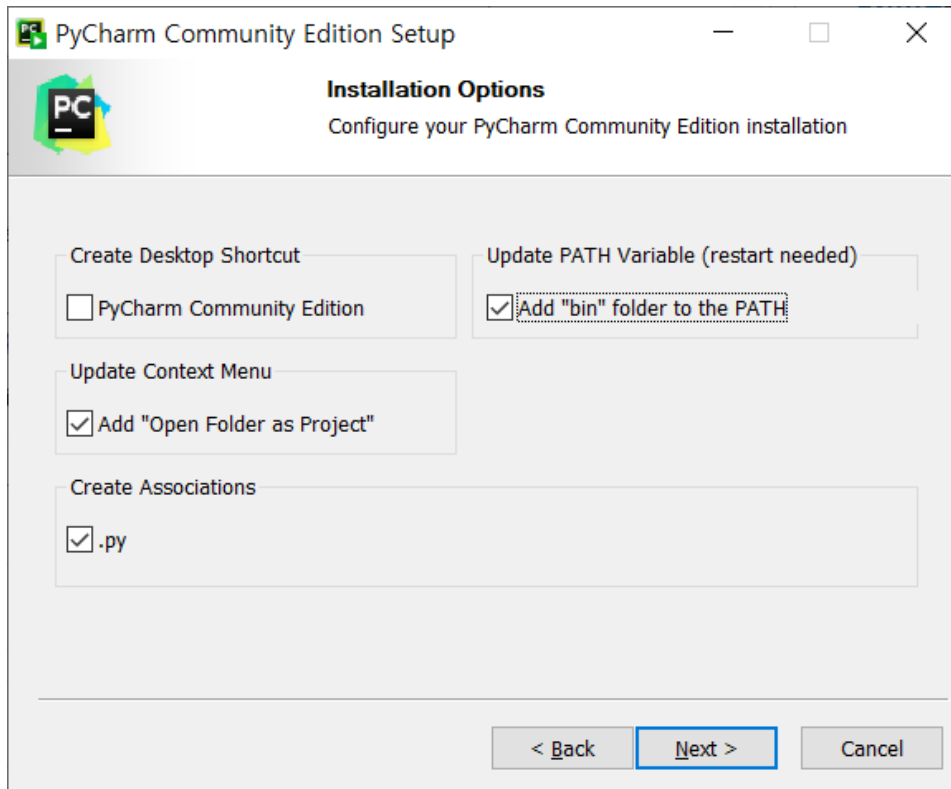
❖ 다운받은 인스톨러를 실행해준다.



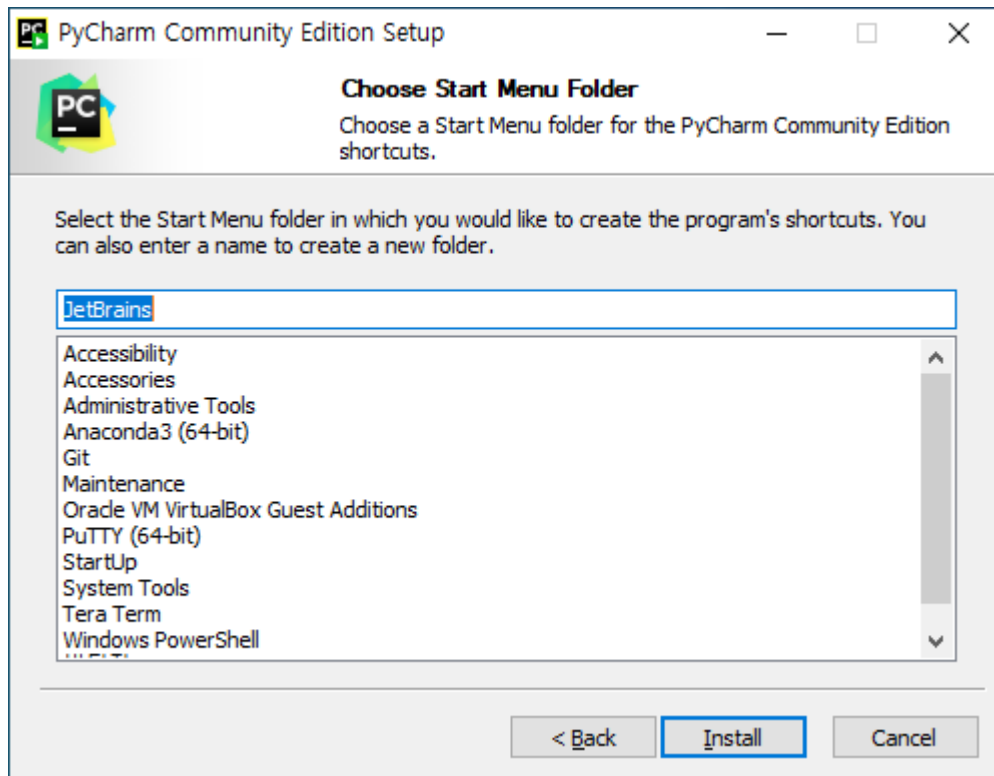
❖ 경로 설정하는 부분 > 기본 값으로 함



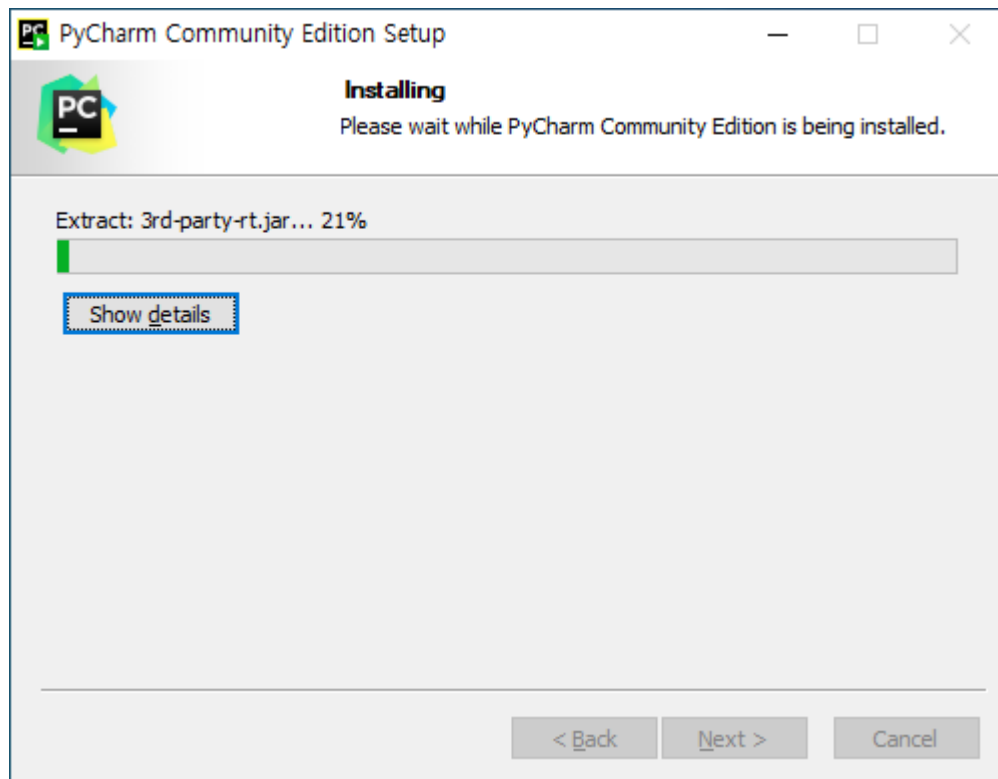
❖ 다운받은 인스톨러 설정 부분



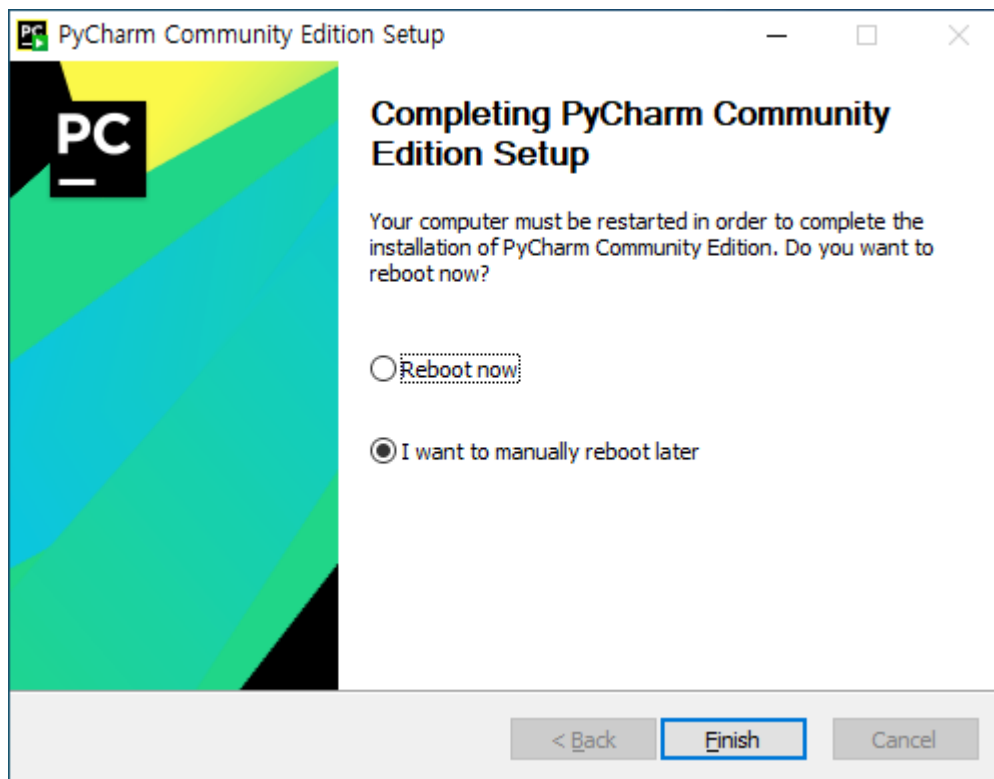
❖ 다운받은 인스톨러 설정 부분



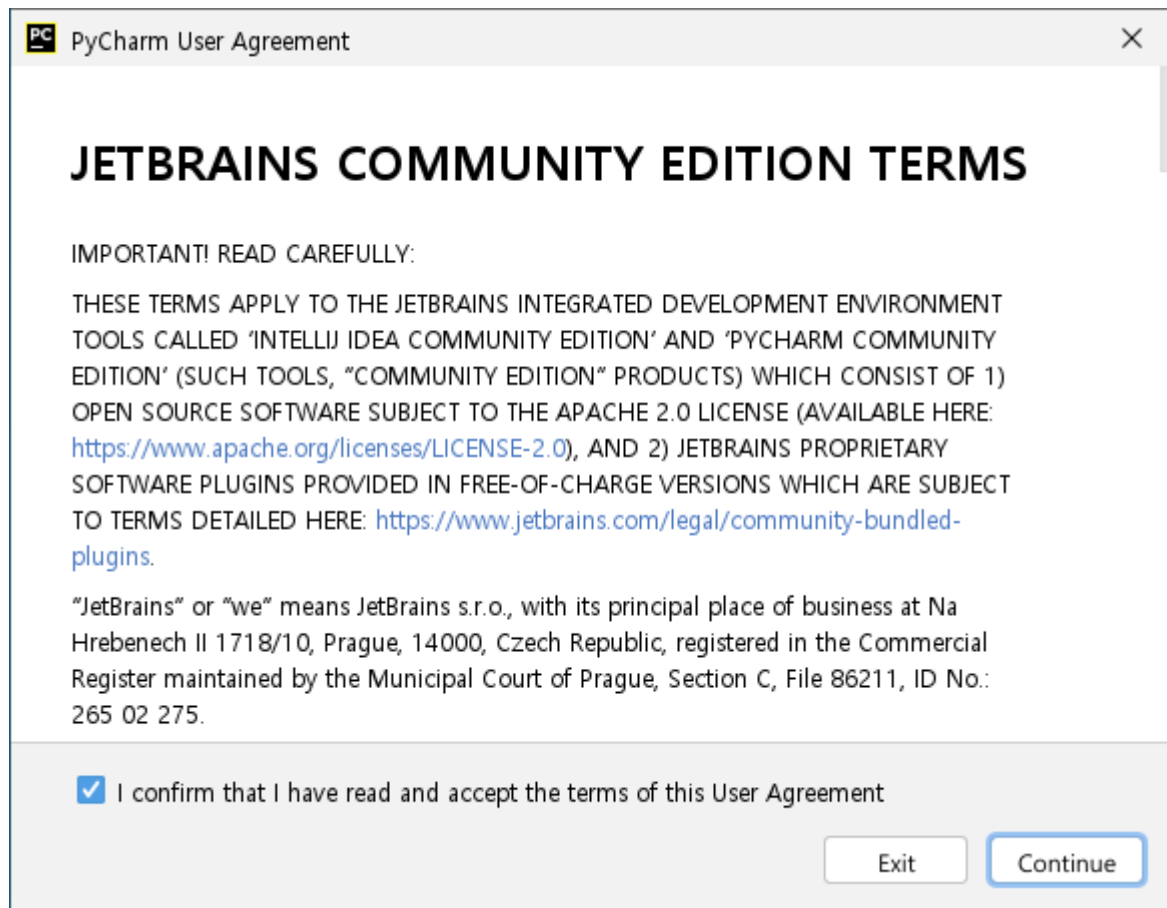
❖ 인스톨 진행



❖ 재부팅 후 PyCharm을 실행해준다.



❖ 동의

A screenshot of the PyCharm User Agreement dialog box. The title bar says "PyCharm User Agreement". The main text is "JETBRAINS COMMUNITY EDITION TERMS". Below that, it says "IMPORTANT! READ CAREFULLY:". The text continues: "THESE TERMS APPLY TO THE JETBRAINS INTEGRATED DEVELOPMENT ENVIRONMENT TOOLS CALLED 'INTELLIJ IDEA COMMUNITY EDITION' AND 'PYCHARM COMMUNITY EDITION' (SUCH TOOLS, 'COMMUNITY EDITION' PRODUCTS) WHICH CONSIST OF 1) OPEN SOURCE SOFTWARE SUBJECT TO THE APACHE 2.0 LICENSE (AVAILABLE HERE: [https://www.apache.org/licenses/LICENSE-2.0](\"https://www.apache.org/licenses/LICENSE-2.0\")), AND 2) JETBRAINS PROPRIETARY SOFTWARE PLUGINS PROVIDED IN FREE-OF-CHARGE VERSIONS WHICH ARE SUBJECT TO TERMS DETAILED HERE: [https://www.jetbrains.com/legal/community-bundled-plugins](\"https://www.jetbrains.com/legal/community-bundled-plugins\")."/>

PC PyCharm User Agreement

JETBRAINS COMMUNITY EDITION TERMS

IMPORTANT! READ CAREFULLY:

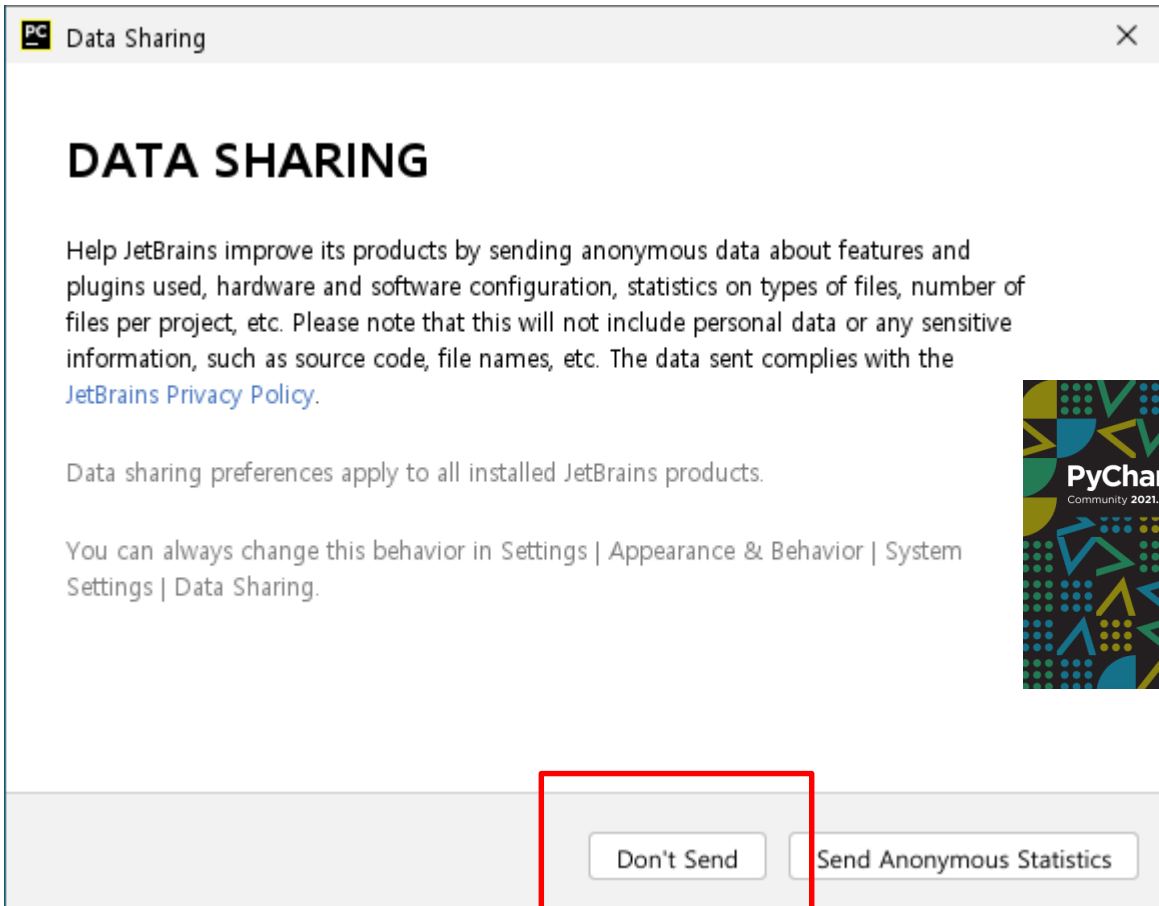
THESE TERMS APPLY TO THE JETBRAINS INTEGRATED DEVELOPMENT ENVIRONMENT TOOLS CALLED 'INTELLIJ IDEA COMMUNITY EDITION' AND 'PYCHARM COMMUNITY EDITION' (SUCH TOOLS, "COMMUNITY EDITION" PRODUCTS) WHICH CONSIST OF 1) OPEN SOURCE SOFTWARE SUBJECT TO THE APACHE 2.0 LICENSE (AVAILABLE HERE: <https://www.apache.org/licenses/LICENSE-2.0>), AND 2) JETBRAINS PROPRIETARY SOFTWARE PLUGINS PROVIDED IN FREE-OF-CHARGE VERSIONS WHICH ARE SUBJECT TO TERMS DETAILED HERE: <https://www.jetbrains.com/legal/community-bundled-plugins>.

"JetBrains" or "we" means JetBrains s.r.o., with its principal place of business at Na Hrebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of Prague, Section C, File 86211, ID No.: 265 02 275.

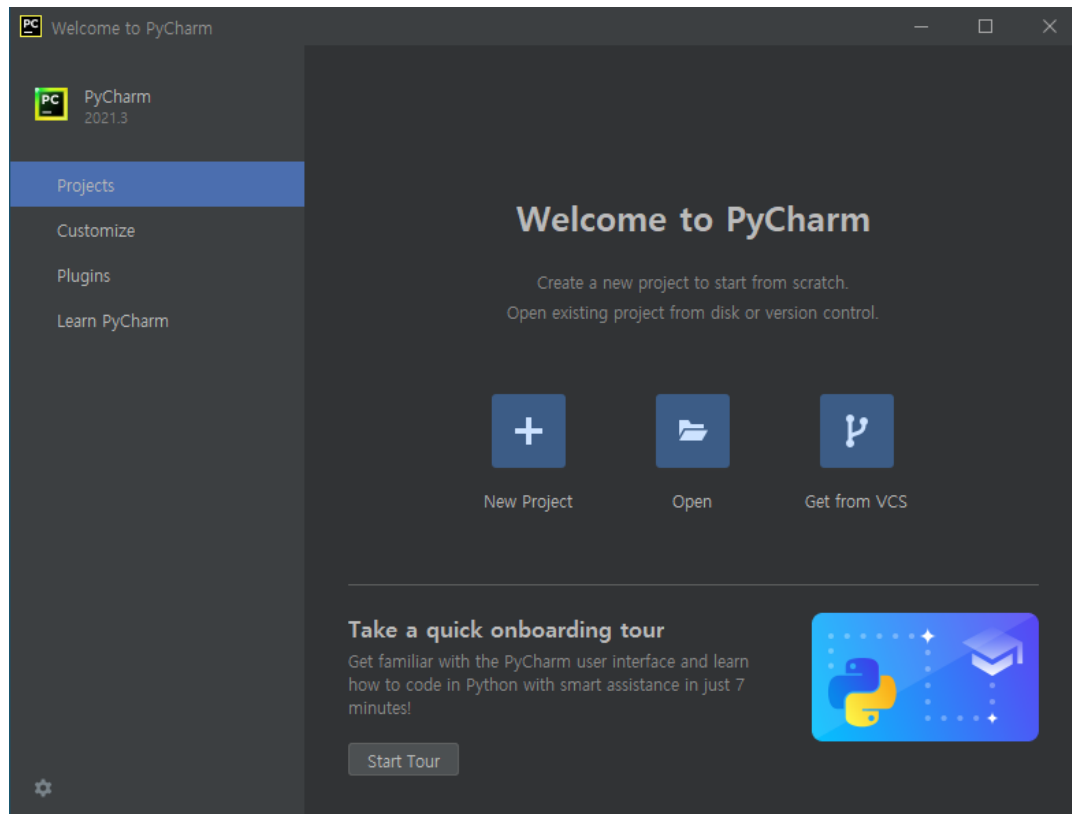
☒ I confirm that I have read and accept the terms of this User Agreement

Exit Continue

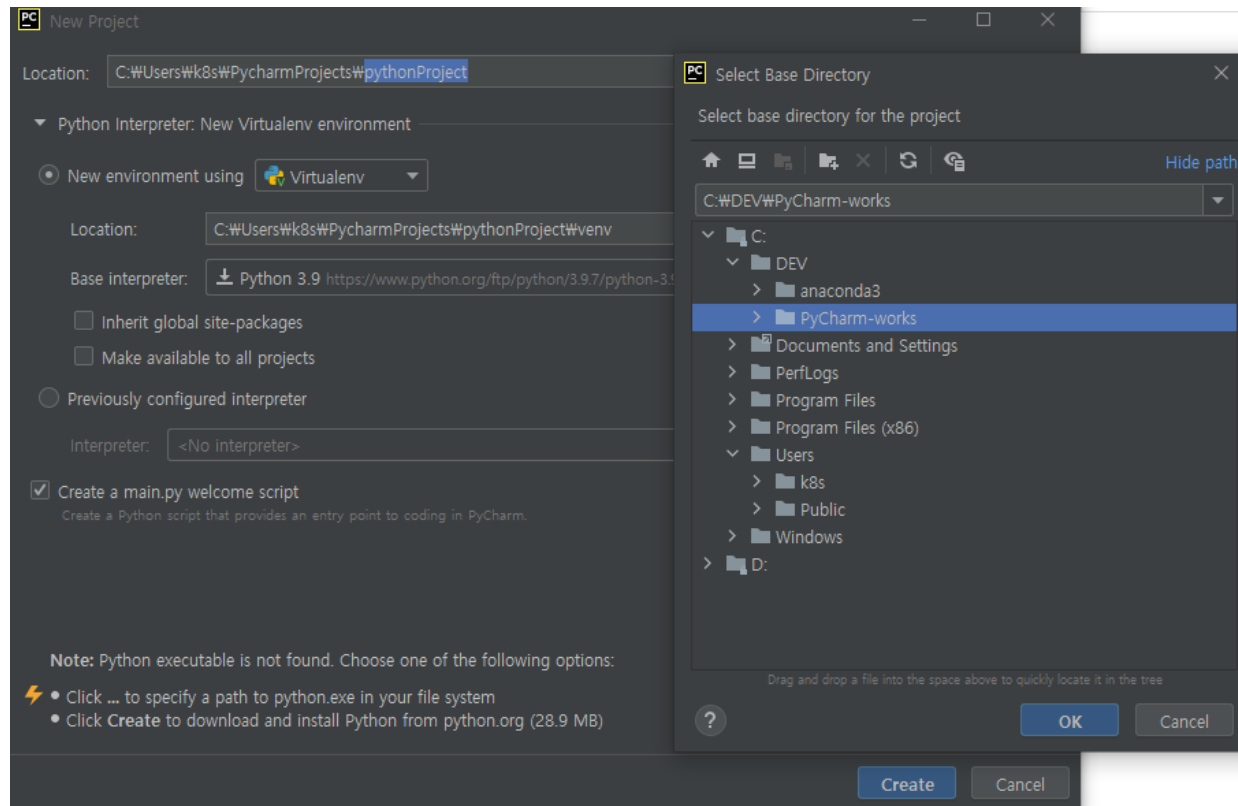
❖ 동의



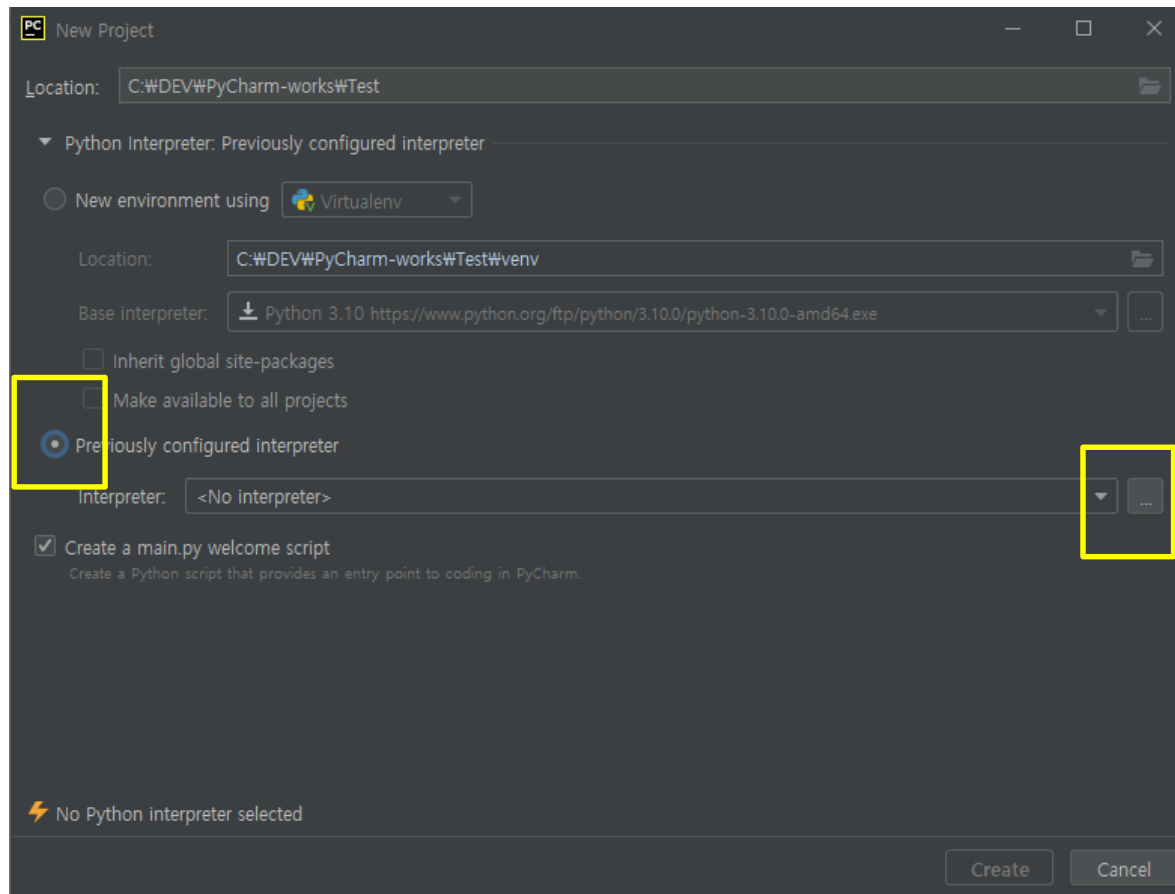
❖ 실행 후 첫 화면 > New Project를 누르고



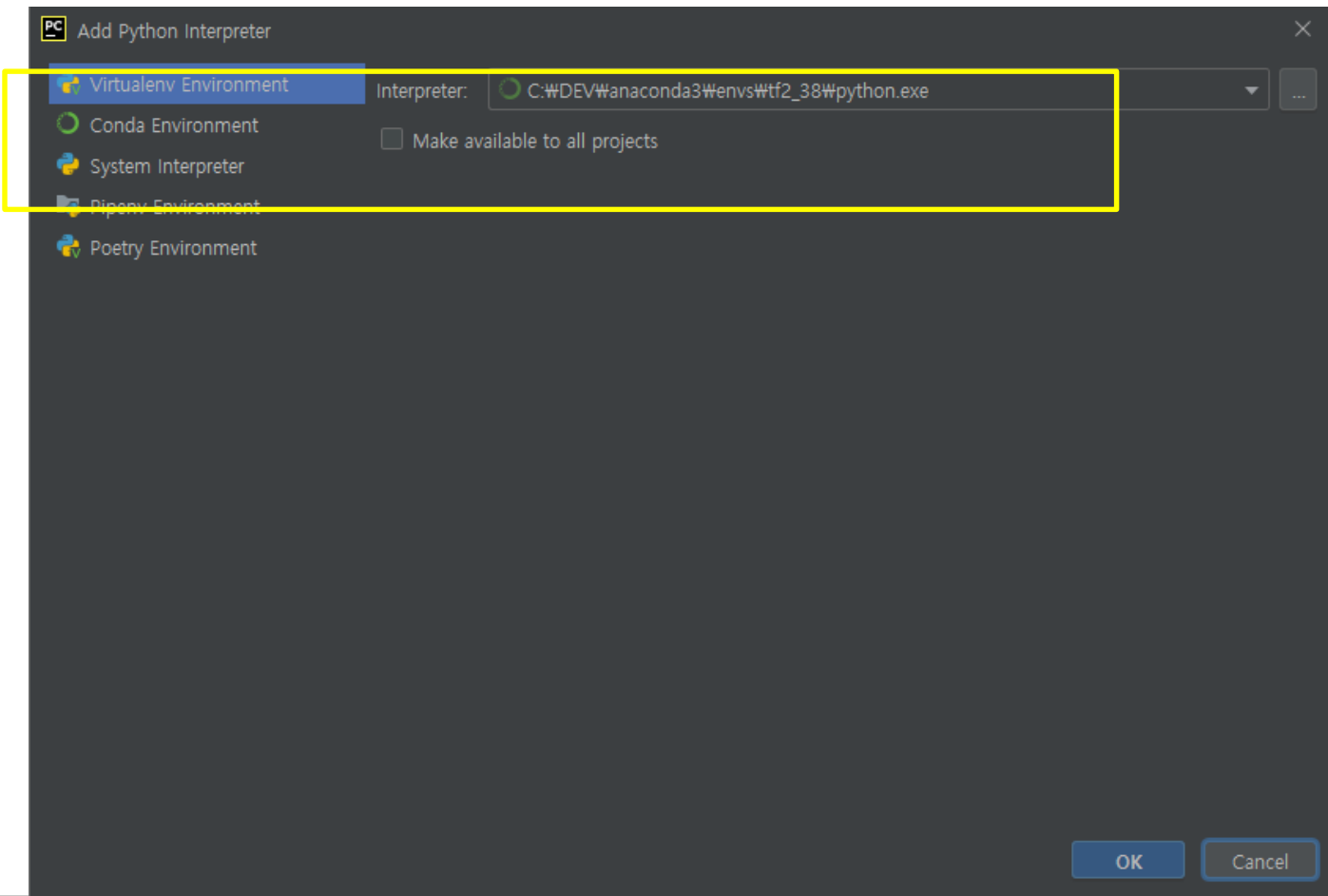
❖ Location에서 작업 경로 설정 > C:\DEV\PyCharm-works



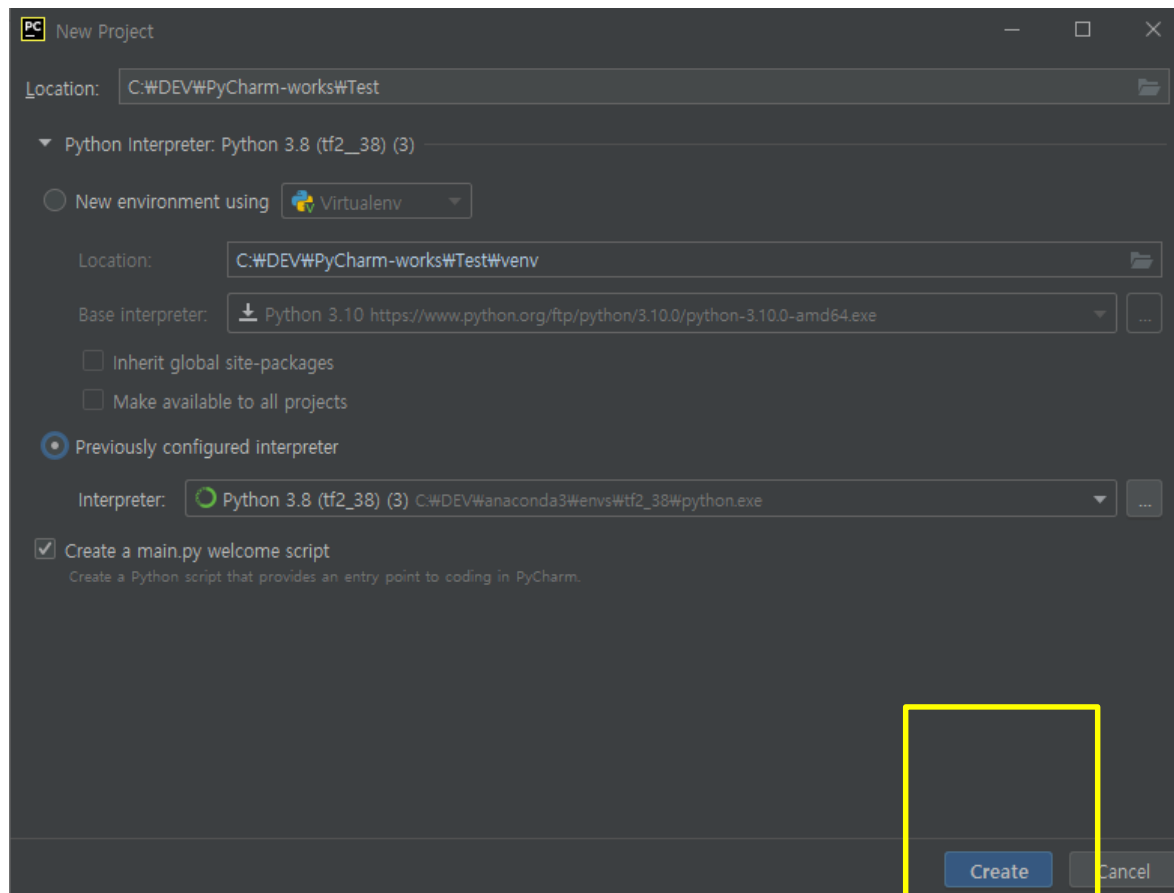
❖ Previously configured interpreter 선택 후 ... 클릭



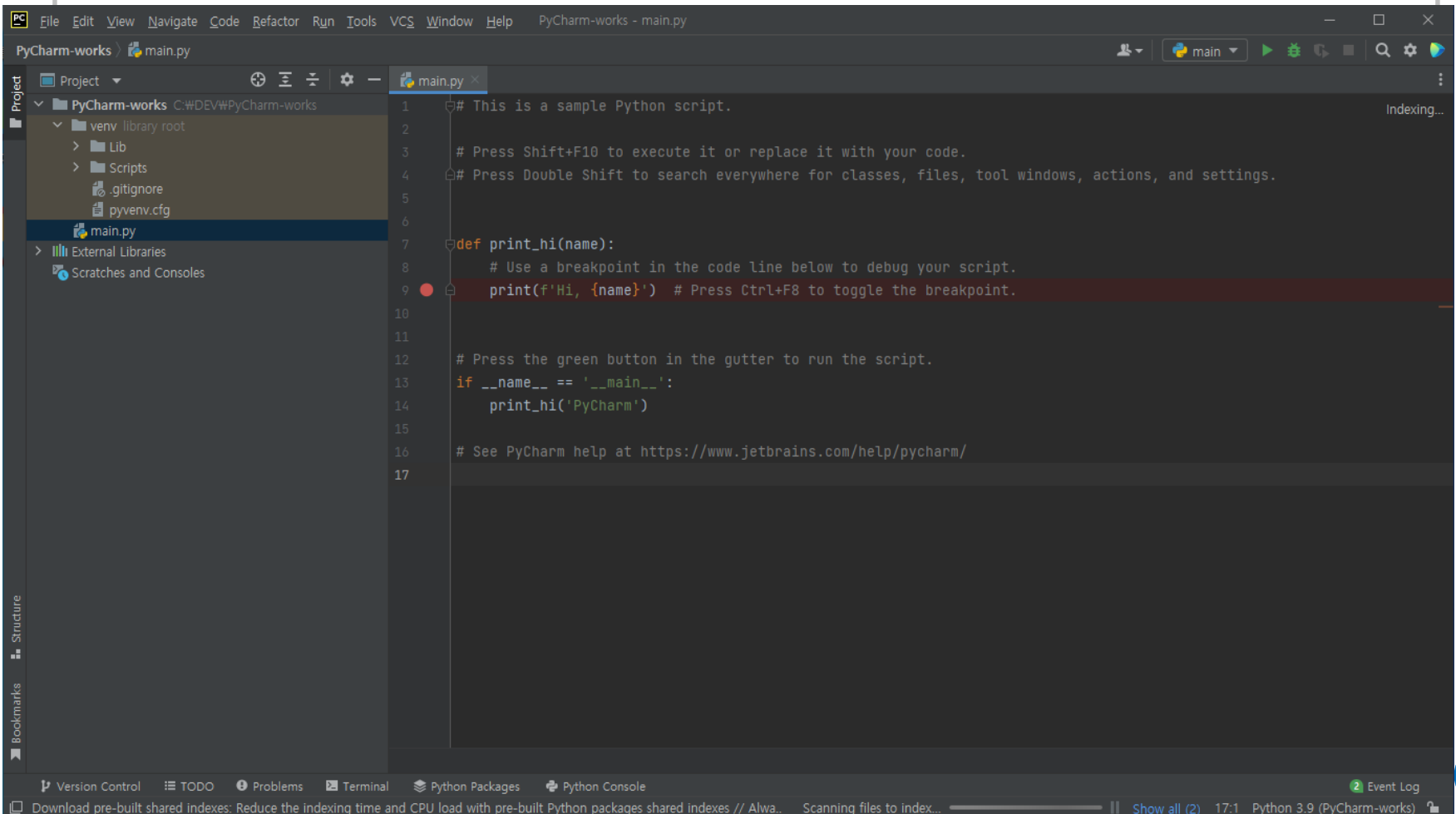
❖ Interpreter 클릭 후 C:\DEV\anaconda3\envs\wf2_38\python.exe 선택



❖ 설정 완료 확인 후 Create



❖ 인터프리터가 제대로 적용되었는지 아나콘다 설치 시 같이 설치된 모듈 numpy로 테스트를 해본다.



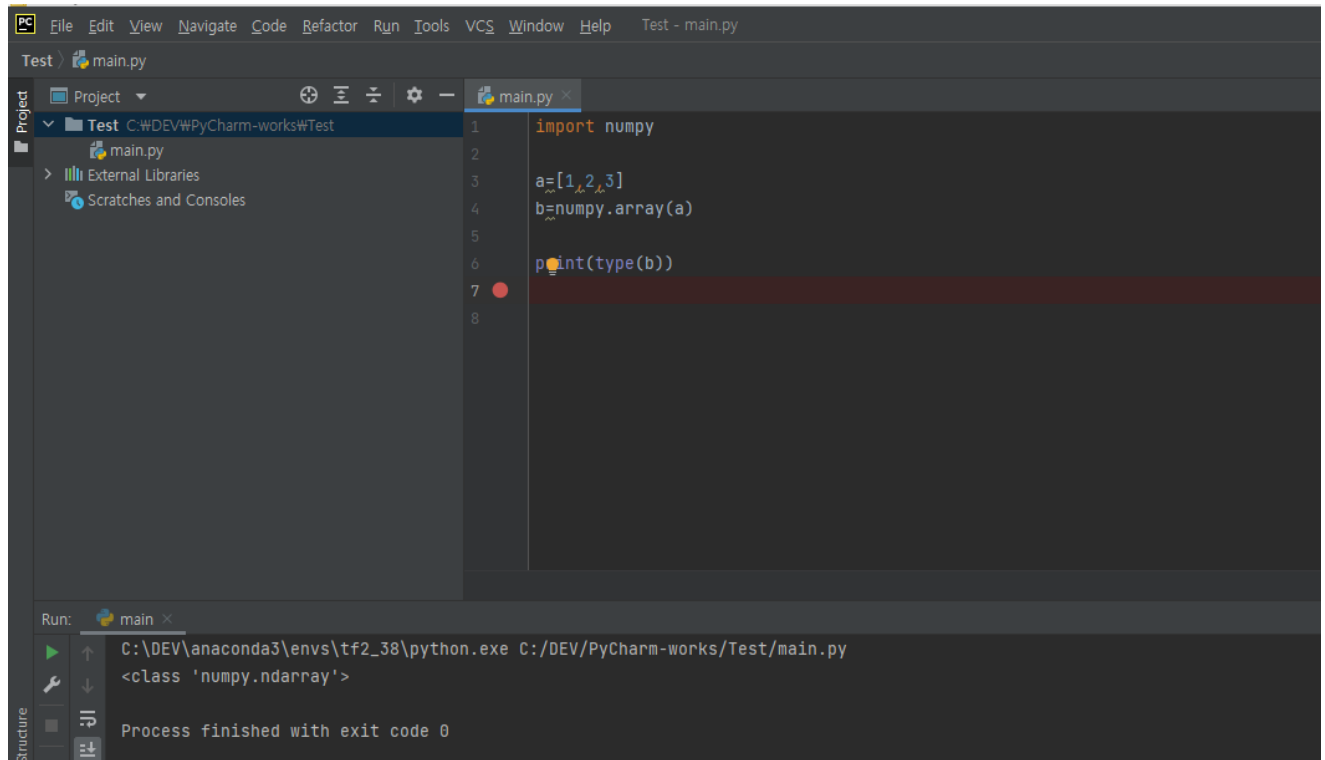
❖ 아래 코드를 작성하고 실행해준다.

```
import numpy
```

```
a=[1,2,3]
```

```
b=numpy.array(a)
```

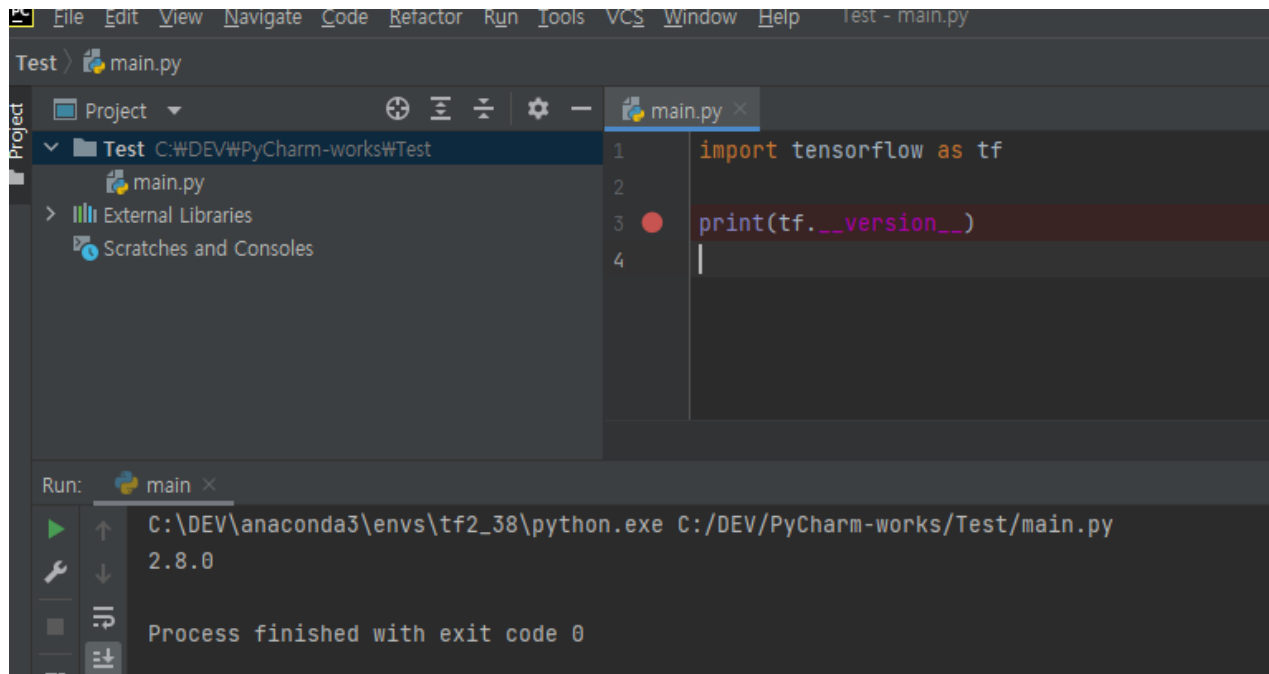
```
print(type(b))
```



❖ 아래 코드를 작성하고 실행해준다.

```
import tensorflow as tf
```

```
print(tf.__version__)
```



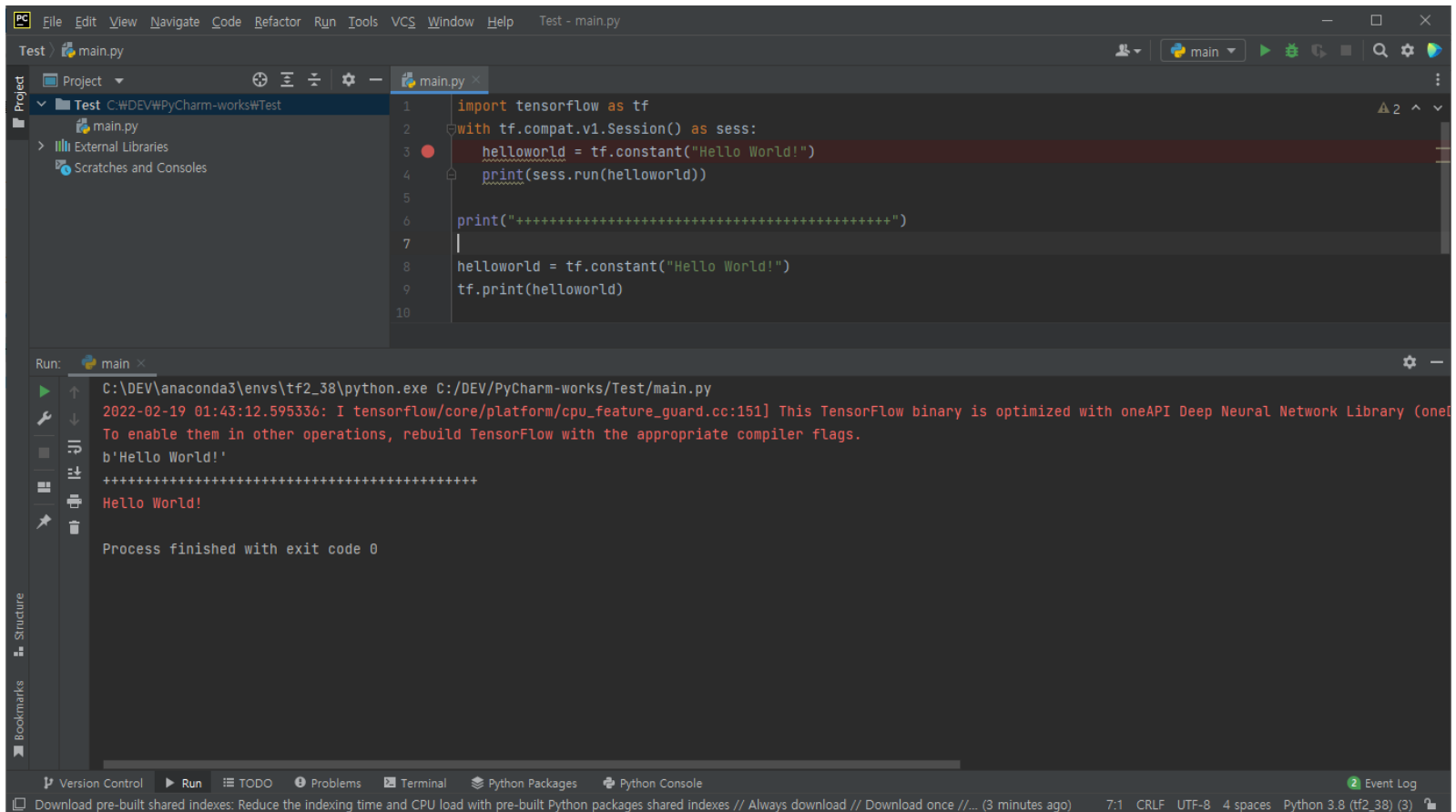
The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The left sidebar shows the Project view with a tree structure: Test (C:\DEV\PyCharm-works\Test) containing main.py, External Libraries, and Scratches and Consoles. The main editor window displays the code from the previous blocks: `import tensorflow as tf` and `print(tf.__version__)`. The Run tab at the bottom shows the execution command: `C:\DEV\anaconda3\envs\tf2_38\python.exe C:/DEV/PyCharm-works/Test/main.py`, the output `2.8.0`, and the status `Process finished with exit code 0`.

❖ 아래 코드를 작성하고 실행해준다.

```
import tensorflow as tf
with tf.compat.v1.Session() as sess:
    helloworld = tf.constant("Hello World!")
    print(sess.run(helloworld))

print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")

helloworld = tf.constant("Hello World!")
tf.print(helloworld)
```



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python file named `main.py` with the following code:

```
1 import tensorflow as tf
2 with tf.compat.v1.Session() as sess:
3     helloworld = tf.constant("Hello World!")
4     print(sess.run(helloworld))
5
6 print("+++++")
7
8 helloworld = tf.constant("Hello World!")
9 tf.print(helloworld)
10
```

The left sidebar shows the project structure with a folder named `Test` containing `main.py`. The bottom panel shows the Run output for the `main` configuration:

```
C:\DEV\anaconda3\envs\tf2_38\python.exe C:/DEV/PyCharm-works/Test/main.py
2022-02-19 01:43:12.595336: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
b'Hello World!'
+++++
Hello World!
Process finished with exit code 0
```

The bottom status bar indicates the current configuration is `main` and shows various tool windows like Version Control, Run, TODO, Problems, Terminal, Python Packages, Python Console, and Event Log.

```
import sys
print("python 버전 : {}".format(sys.version))

import pandas as pd
print("pandas 버전 : {}".format(pd.__version__))

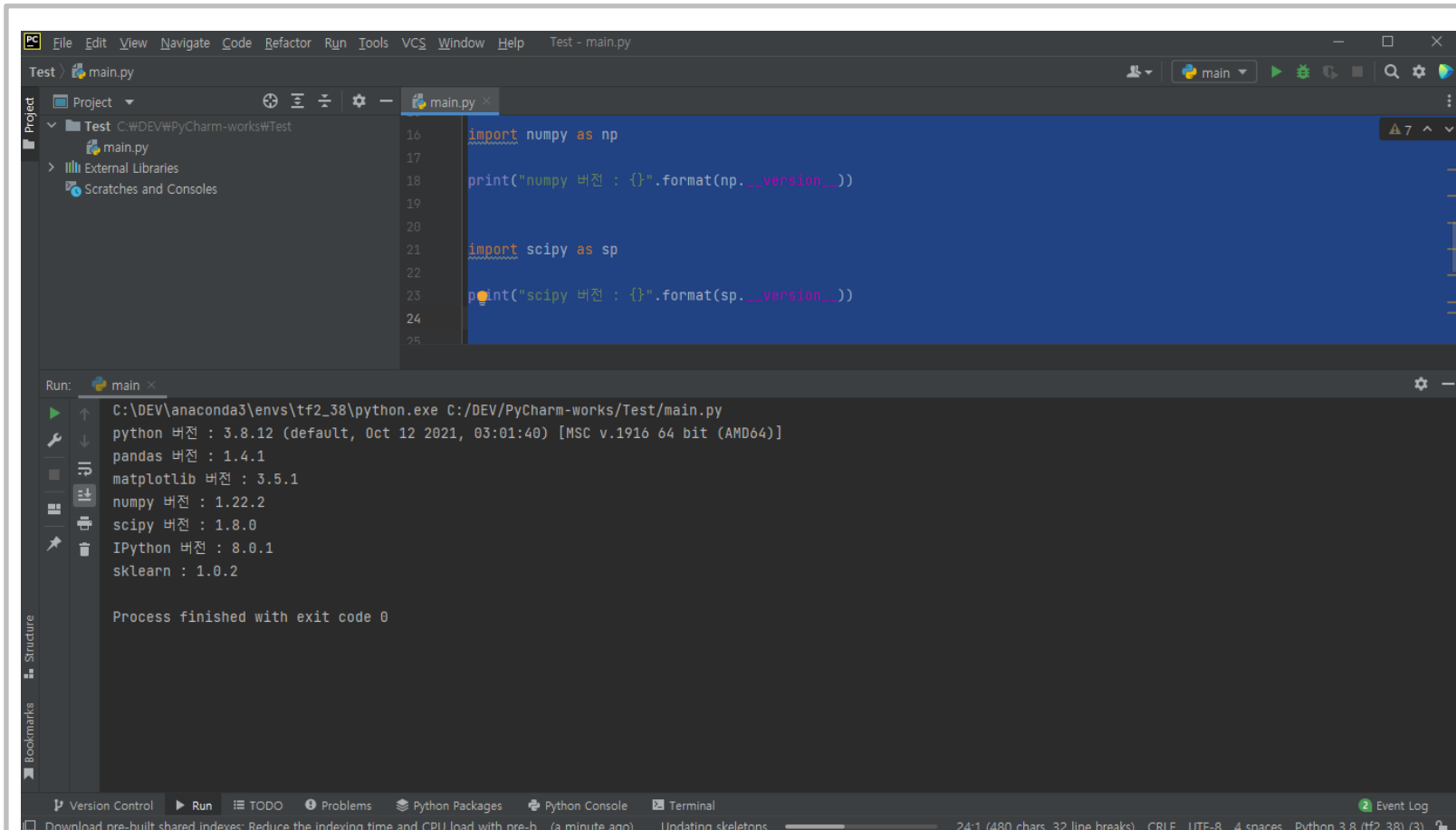
import matplotlib
print("matplotlib 버전 : {}".format(matplotlib.__version__))

import numpy as np
print("numpy 버전 : {}".format(np.__version__))

import scipy as sp
print("scipy 버전 : {}".format(sp.__version__))

import IPython
print("IPython 버전 : {}".format(IPython.__version__))

import sklearn
print("sklearn : {}".format(sklearn.__version__))
```



The image shows a PyCharm IDE window with a project named 'Test'. The main editor displays a Python script named 'main.py' with the following code:

```
16 import numpy as np
17
18 print("numpy 버전 : {}".format(np.__version__))
19
20
21 import scipy as sp
22
23 print("scipy 버전 : {}".format(sp.__version__))
24
25
```

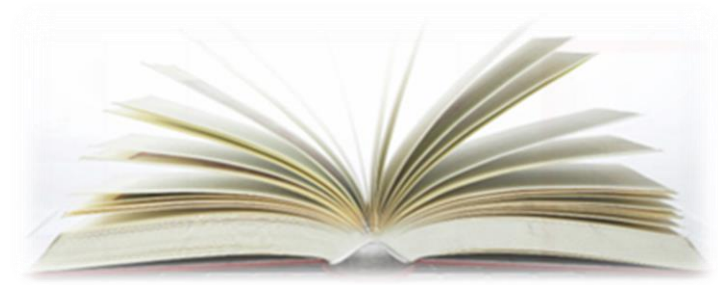
The Run console at the bottom shows the output of the script:

```
Run: main
C:\DEV\anaconda3\envs\tf2_38\python.exe C:/DEV/PyCharm-works/Test/main.py
python 버전 : 3.8.12 (default, Oct 12 2021, 03:01:40) [MSC v.1916 64 bit (AMD64)]
pandas 버전 : 1.4.1
matplotlib 버전 : 3.5.1
numpy 버전 : 1.22.2
scipy 버전 : 1.8.0
IPython 버전 : 8.0.1
sklearn : 1.0.2

Process finished with exit code 0
```

The bottom status bar indicates the file encoding is UTF-8 and the Python interpreter is 3.8 (tf2_38).

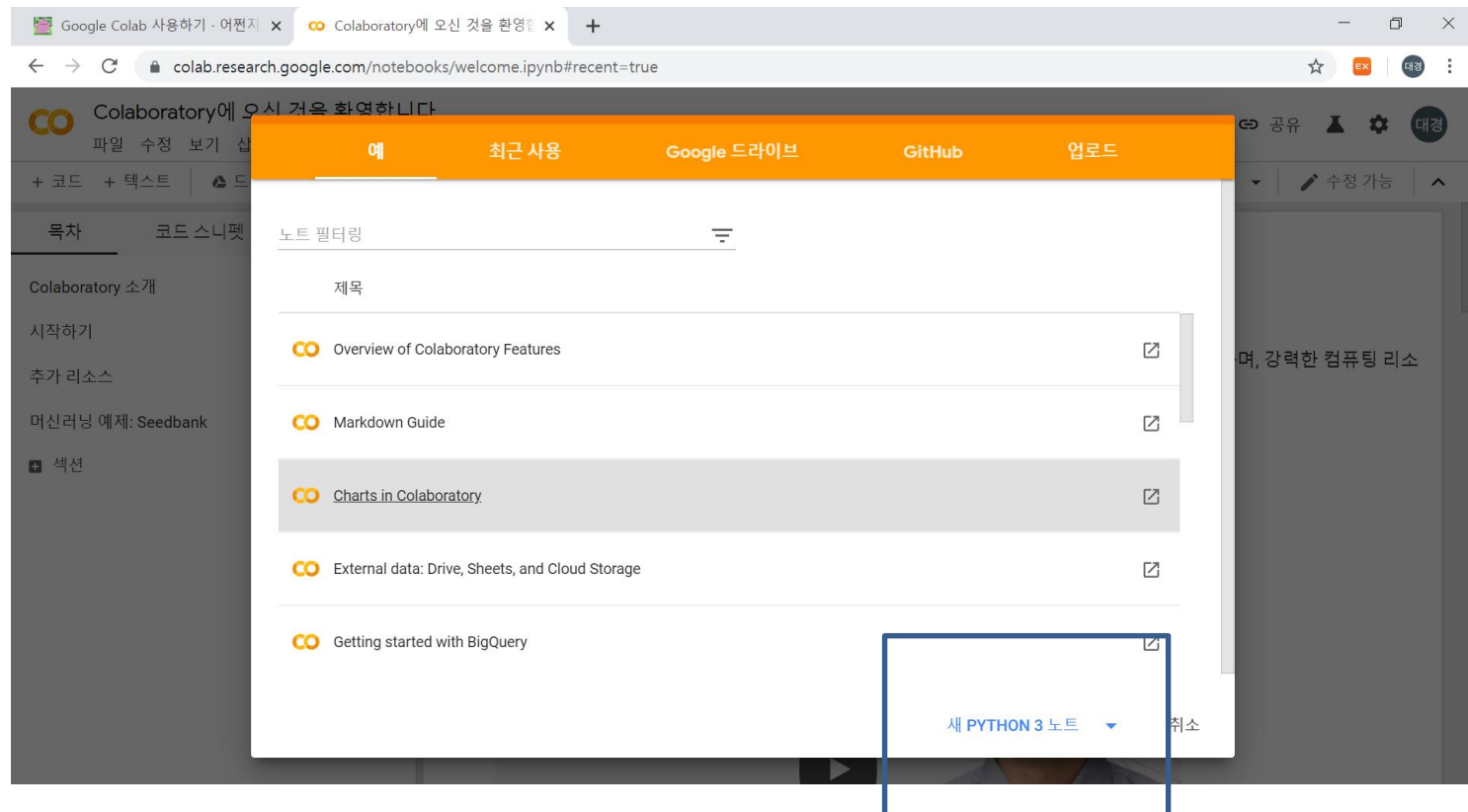
Unit 2



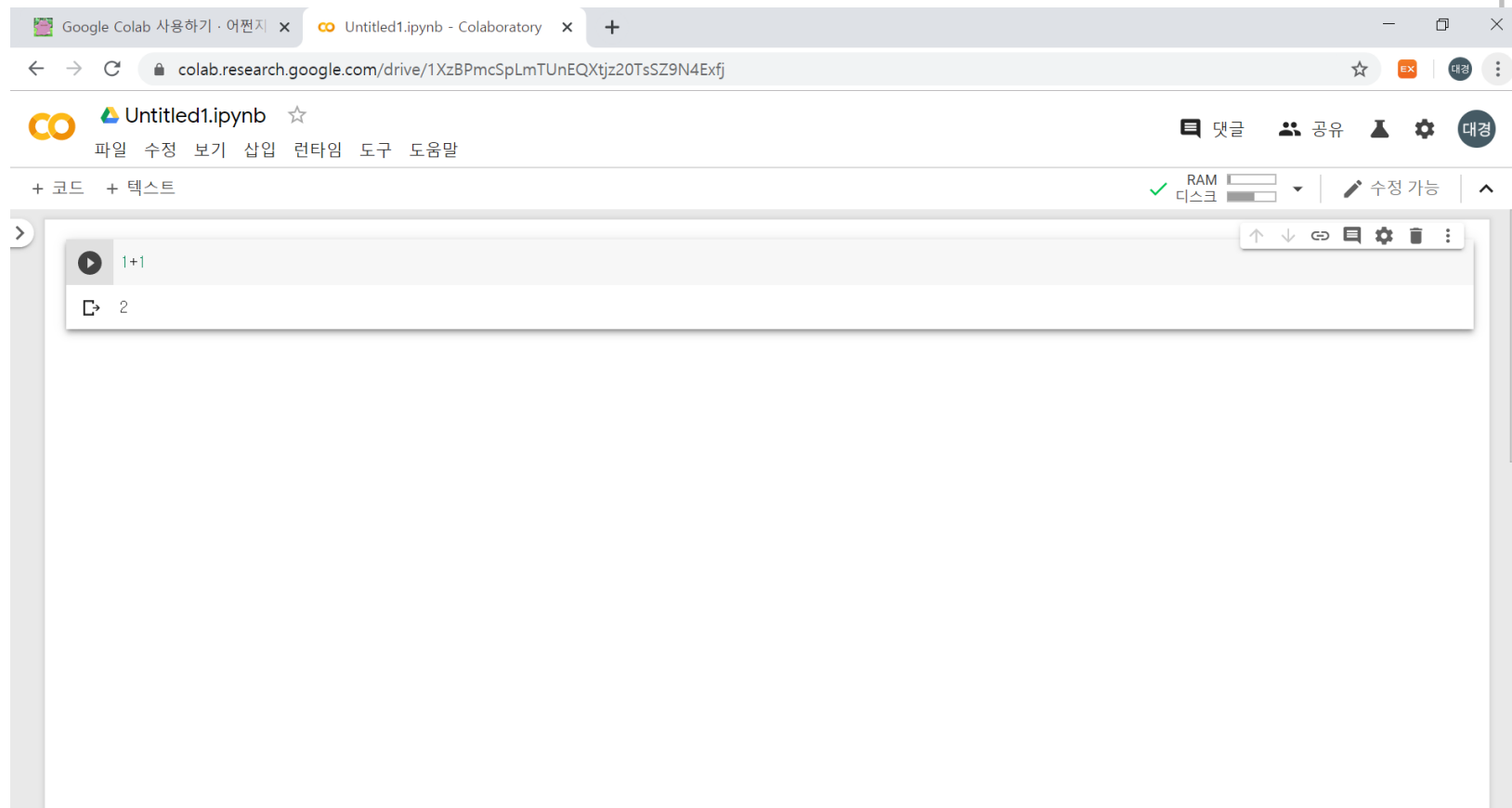
Google Colab

- ❖ Google Colab은 AI개발자들을 위해 구글에서 제공하는 무료 클라우드 서비스
- ❖ 풀 네임은 Google Colaboratory
- ❖ Google Drive + Jupyter Notebook
 - Google Drive처럼 협업 가능(동시에 수정 가능)
- ❖ <https://colab.research.google.com/>로 접속시 사용 가능
- ❖ 컴퓨터 사양(19년 12월 기준)
 - Ubuntu 17.10
 - CPU 제논 2.3GHz
 - 메모리 13G
 - GPU : K80 또는 T4 :
 - TPU도 사용 가능
- ❖ GPU 사용시 최대 12시간
- ❖ Github의 소스 코드를 Colab에서 사용 가능

❖ <https://colab.research.google.com/>로 접속시 사용 가능



❖ Python3로 편집기 실행



❖ 붉은 부분

❖ 해당 노트북의 목차

❖ 코드 스니펫

❖ 재사용 가능한 소스 코드로 다양한 예제 코드가 있음

❖ Uploading files from your local file system, Using BigQuery, Listing files in Google Drive, Install library, etc

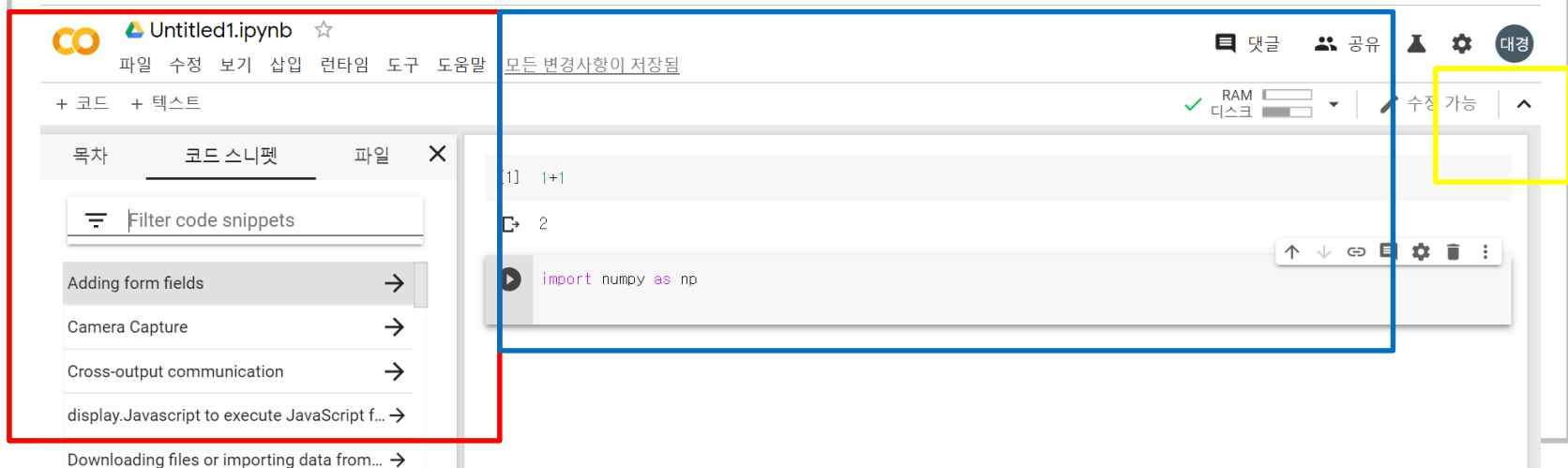
❖ 연결된 파일

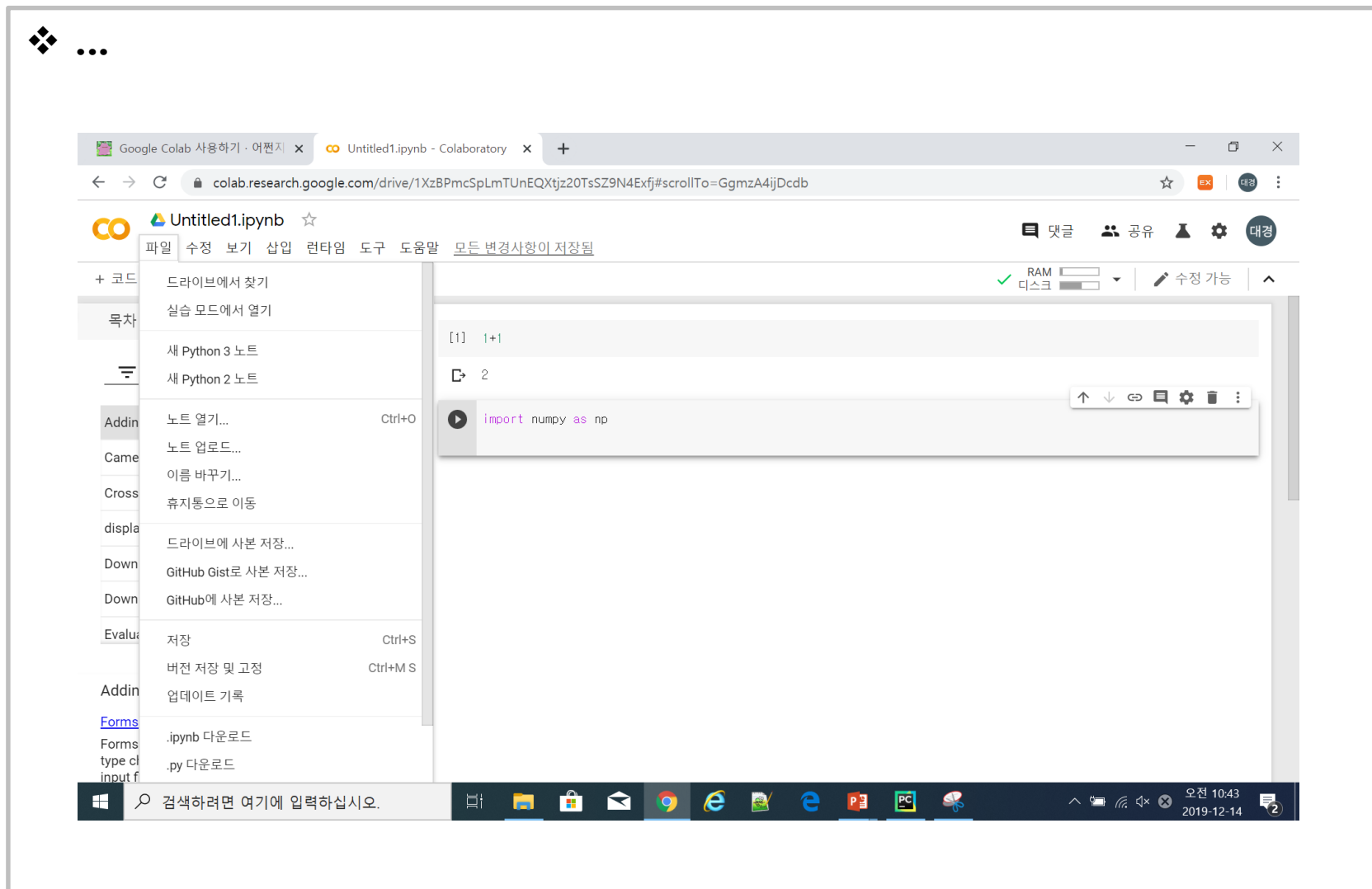
❖ 노란 부분

❖ 헤더 보이게 유무

❖ 파란 부분

❖ 코드 부분





❖ 다음 명령어를 입력

```
import numpy

a=[1,2,3]
b=numpy.array(a)

print(type(b))
```

```
<class 'numpy.ndarray'>
```

```
import tensorflow as tf

print(tf.__version__)
```

```
2.8.0
```

```
import tensorflow as tf
with tf.compat.v1.Session() as sess:
    helloworld = tf.constant("Hello World!")
    print(sess.run(helloworld))

print("+++++")
```

```
helloworld = tf.constant("Hello World!")
tf.print(helloworld)
```

```
b'Hello World!'
+++++
Hello World!
```

❖ 다음 명령어를 입력

파이썬 라이브러리

```
import sys

print("python 버전 : {}".format(sys.version))

import pandas as pd

print("pandas 버전 : {}".format(pd.__version__))

import matplotlib

print("matplotlib 버전 : {}".format(matplotlib.__version__))

import numpy as np

print("numpy 버전 : {}".format(np.__version__))

import scipy as sp

print("scipy 버전 : {}".format(sp.__version__))

import IPython

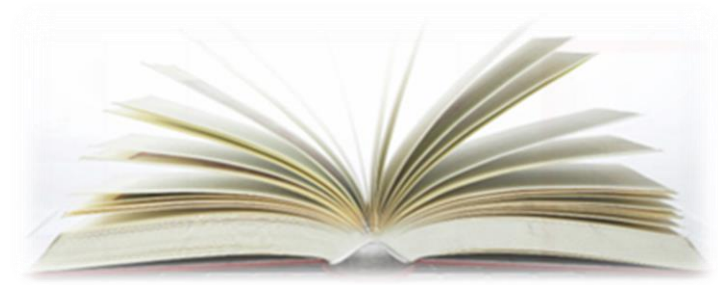
print("IPython 버전 : {}".format(IPython.__version__))

import sklearn

print("sklearn : {}".format(sklearn.__version__))
```

```
python 버전 : 3.7.12 (default, Jan 15 2022, 18:48:18)
[GCC 7.5.0]
pandas 버전 : 1.3.5
matplotlib 버전 : 3.2.2
numpy 버전 : 1.21.5
scipy 버전 : 1.4.1
IPython 버전 : 5.5.0
sklearn : 1.0.2
```

Unit A



참고자료

- ❖ <http://www.ncs.go.kr>
- ❖ NELDALE/JOHN LEWIS 지음, 조영석/김대경/박찬영/송창근 역, 단계별로 배우는 컴퓨터과학, 홍릉과학출판사, 2018
- ❖ 혼자 공부하는 머신러닝+딥러닝 박해선 지음 | 한빛미디어 | 2020년 12월
- ❖ 머신러닝 실무 프로젝트 ,아리가 미치아키, 나카야마 신타, 니시바야시 다카시 지음 | 심효섭 옮김 | 한빛미디어 | 2018년 06월
- ❖ 파이썬을 활용한 머신러닝 쿡북 크리스 알본 지음 | 박해선 옮김 | 한빛미디어 | 2019년 09월
- ❖ 처음 배우는 머신러닝 김의중 지음 | 위키북스 | 2016년 07월
- ❖ 파이썬으로 배우는 머신러닝의 교과서 : 이토 마코토 지음 | 박광수(아크몬드) 옮김 | 한빛미디어 | 2018년 11월
- ❖ 기타 서적 및 웹 사이트 자료 다수 참조



감사합니다.

- ❖ Mobile: 010-9591-1401
- ❖ E-mail: onlooker2zip@naver.com