



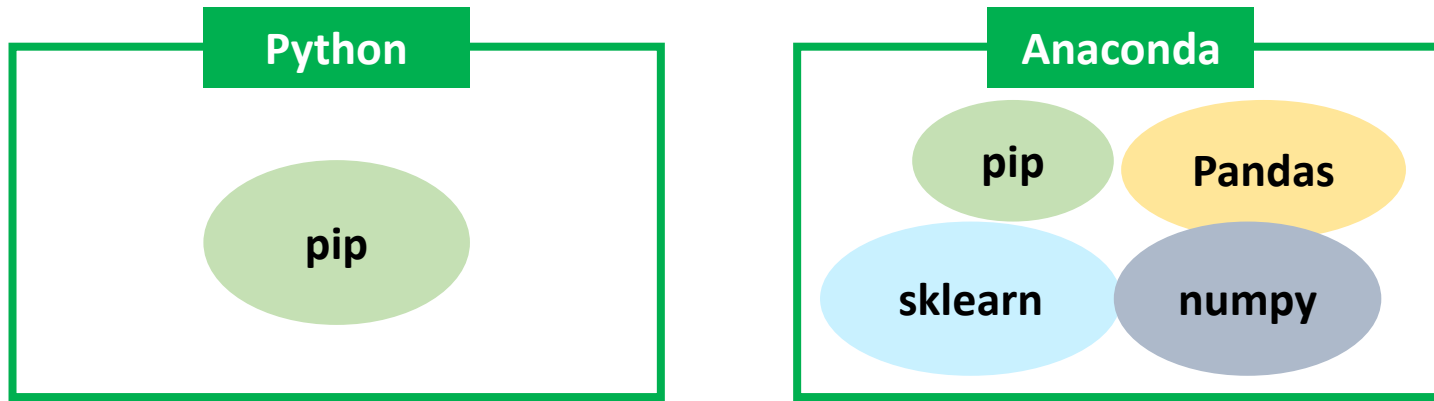
K-Digital Training 스마트 팩토리 3기

# Anaconda

# Anaconda

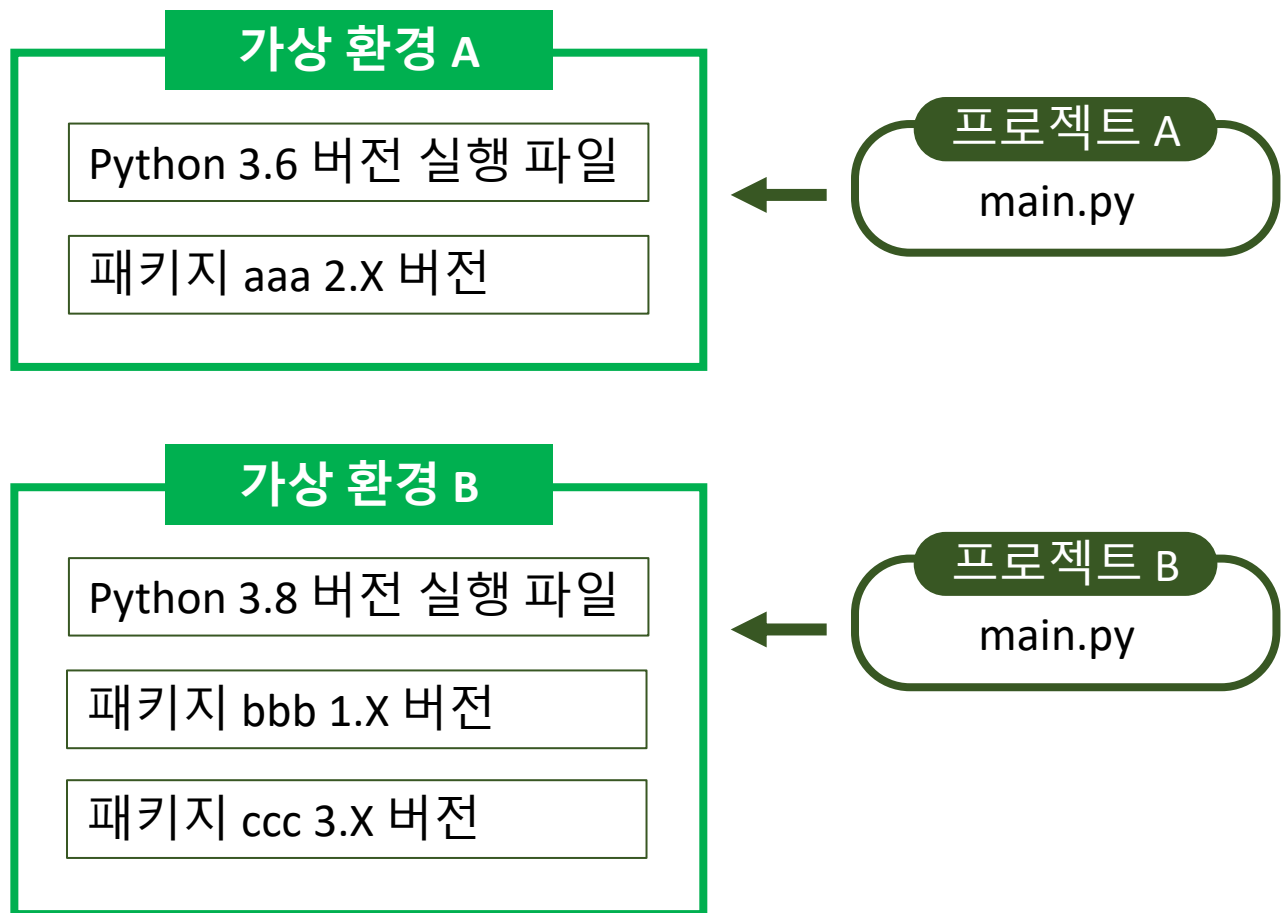
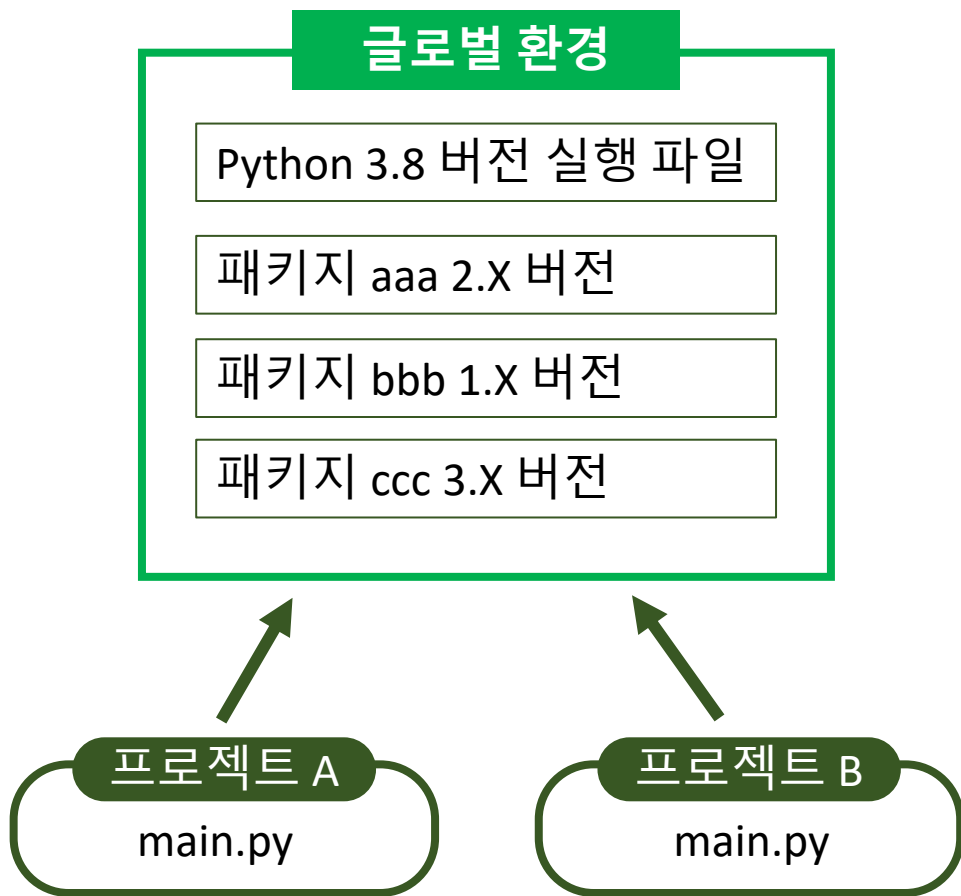
- 데이터 분석, 머신 러닝 등에 사용하는 여러가지 패키지가 기본적으로 포함 되어있는 파이썬
- 파이썬 가상 환경을 구축하는데도 유용하게 사용
- 내부적으로 conda라는 환경/패키지 관리자가 존재
- conda를 통해 패키지를 설치하거나 가상 환경을 관리할 수 있음

# Python vs Anaconda





- **Python**은 기본적으로 패키지 관리 시스템인 pip만을 포함하고 있음
  - ➡ 필요한 툴, 패키지가 있다면 pip을 통해 수동으로 추가해야 함.
  - ➡ 패키지가 컴퓨터 자체에 설치됨
  - ➡ 프로젝트를 여러 번 진행하다 보면 필요한 패키지는 2~3개 정도면 되는데, 10개 15개의 패키지들이 설치되어 필요 이상으로 공간을 차지하기도 함.
- **Anaconda**는 데이터 분석, 머신 러닝 등에 사용하는 여러가지 패키지가 기본적으로 포함되어 있음

# 가상 환경 vs 글로벌 환경



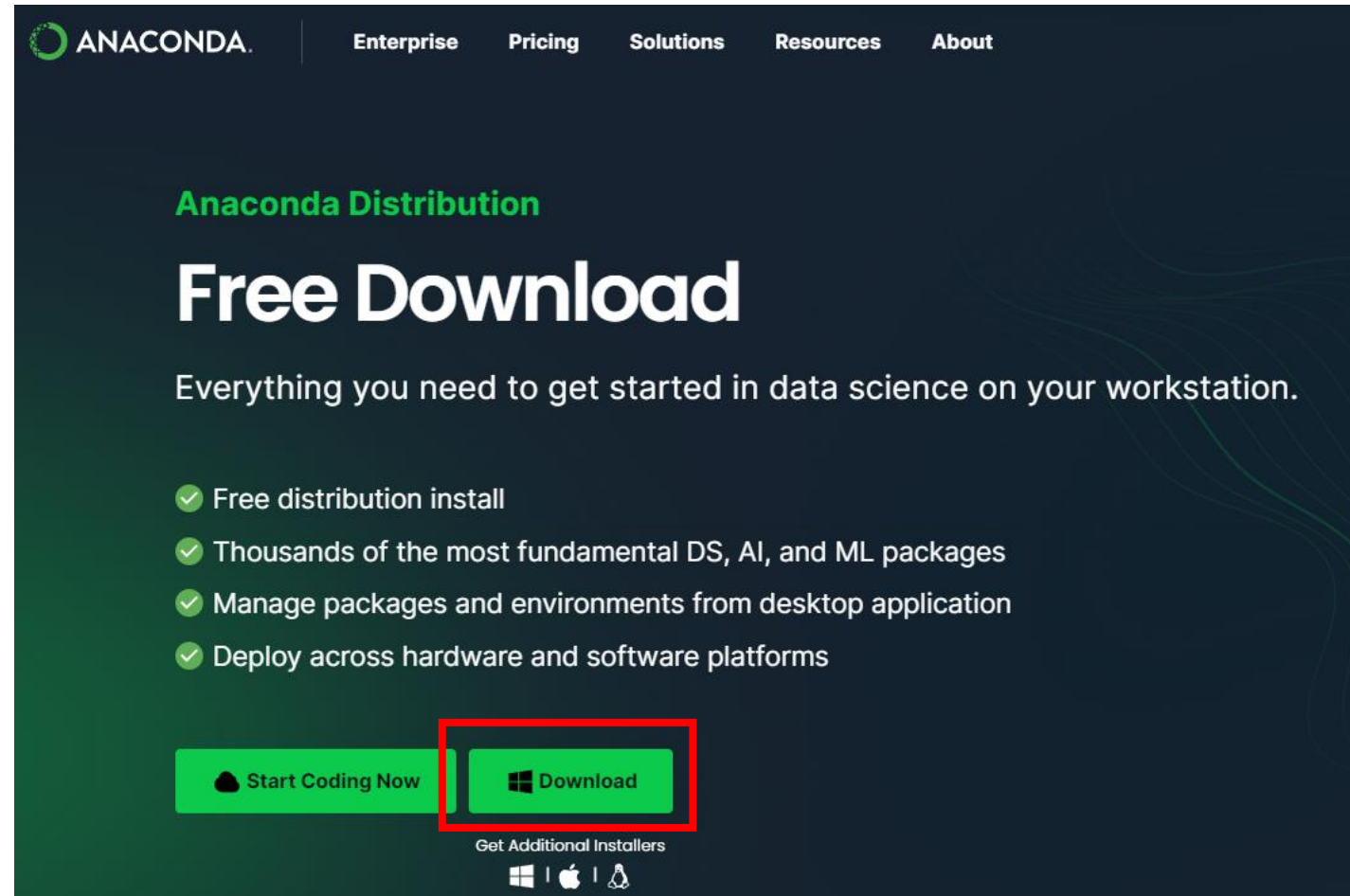
# 가상환경이 필요한 이유?

- 프로젝트를 진행하다 보면 여러 library, package를 다운로드 하게  
되  
 각 library, package들끼리 **충돌**을 일으키는 경우 **↑**
-  이런 경우에 가상환경을 이용한다면?  
프로젝트별로 **독립적인 작업 환경**에서 작업할 수 있다!

# Anaconda 설치하기

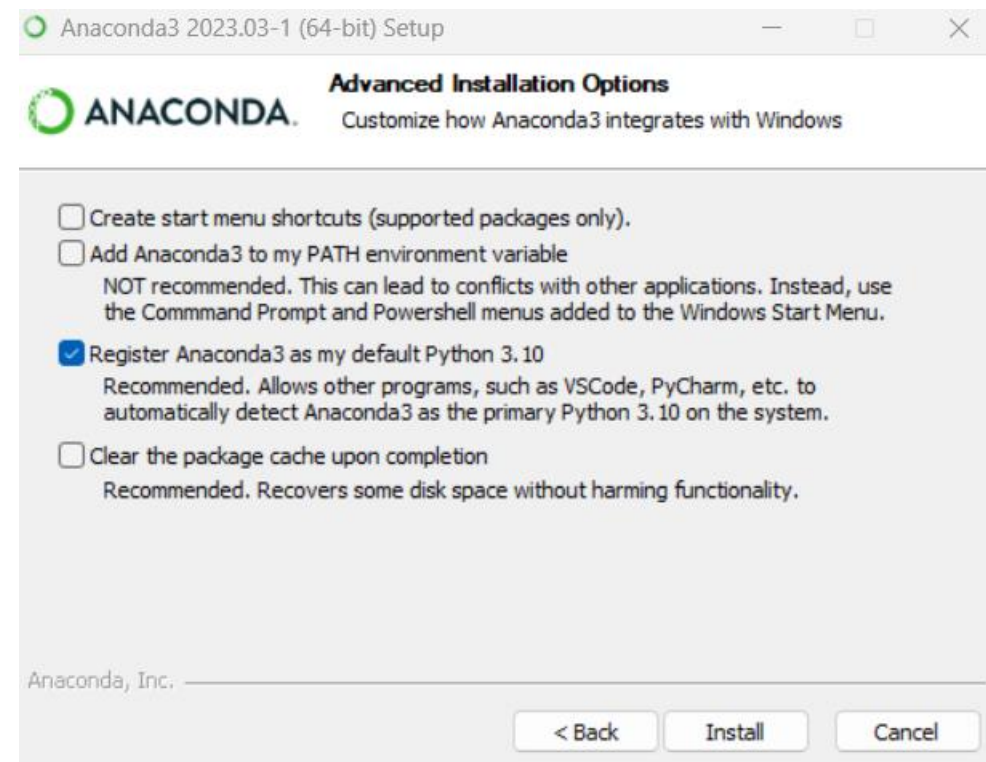
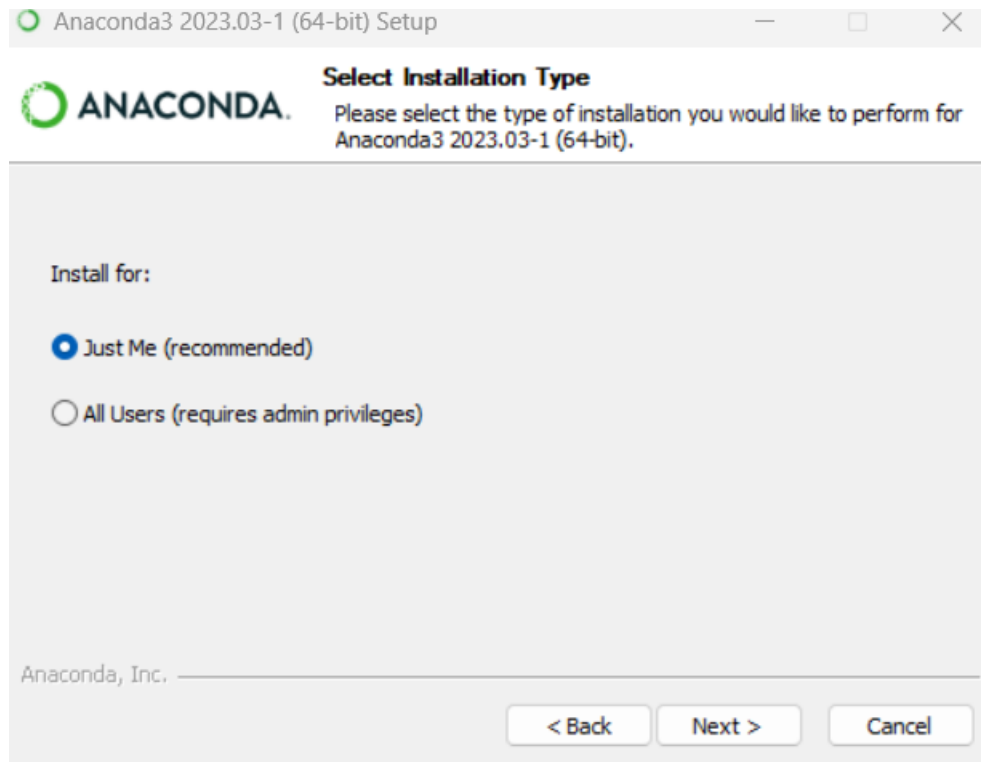
# Anaconda 설치

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

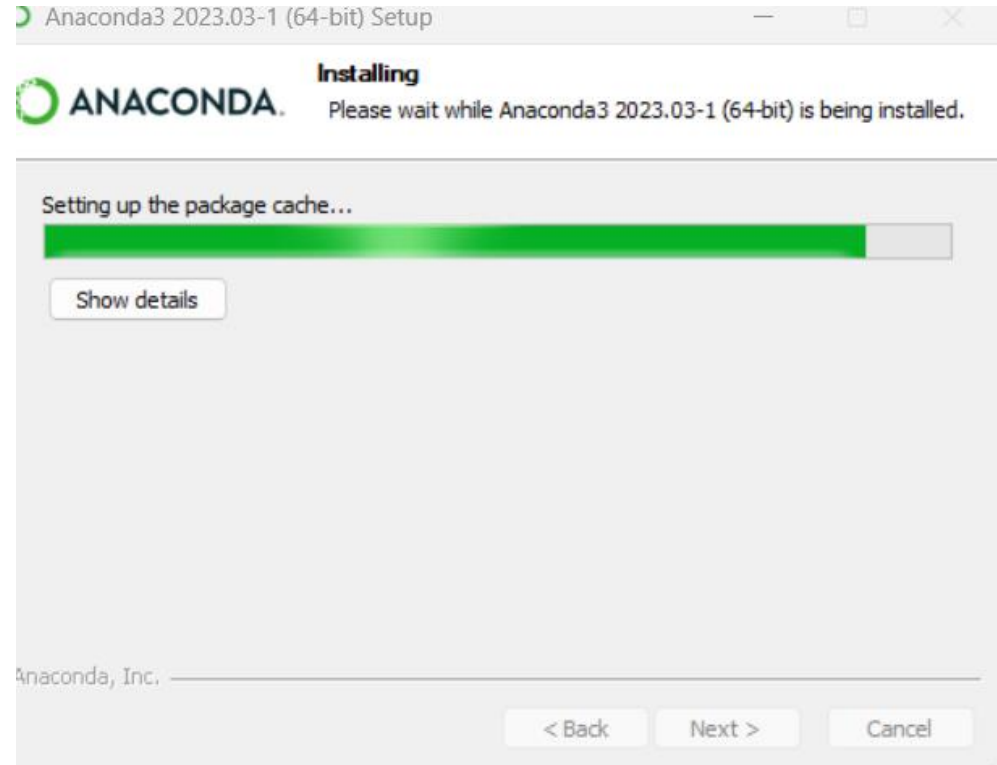
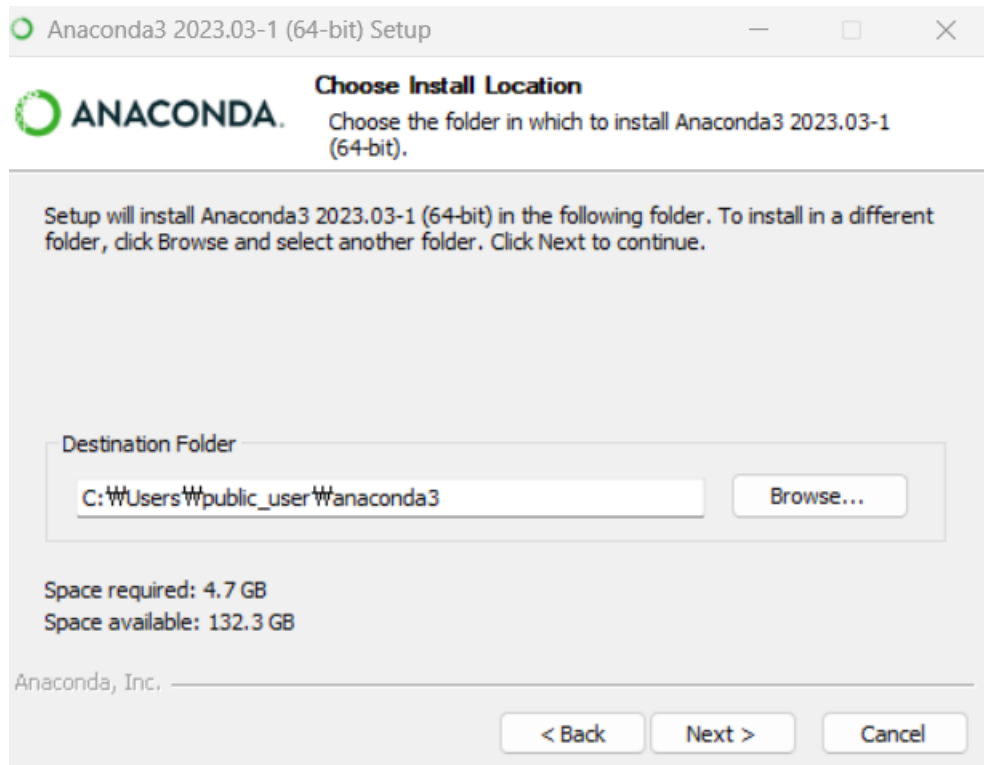




# Anaconda 설치



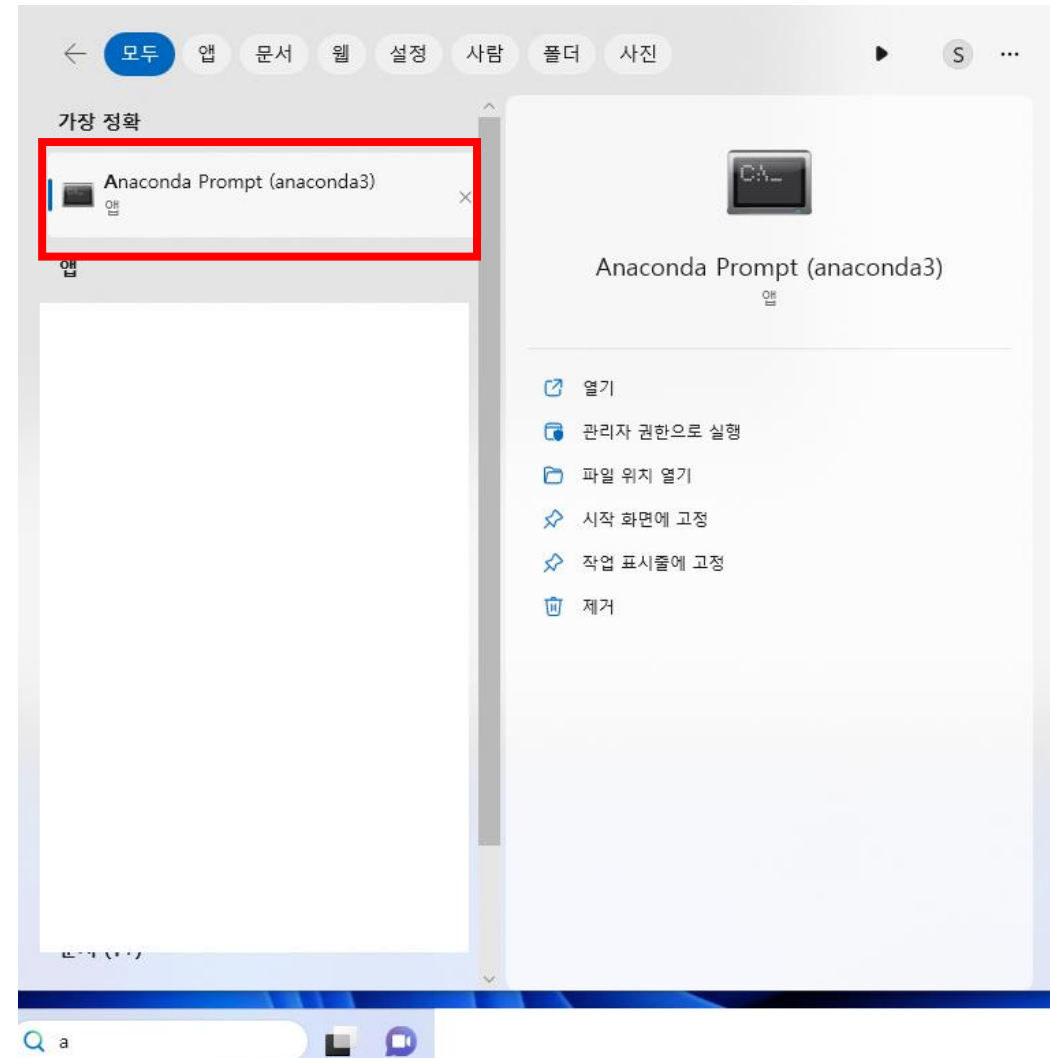
# Anaconda 설치



⇒ 시간이 다소 소요됨

# Anaconda 가상 환경 관리

- Anaconda Prompt 실행



# Anaconda 가상 환경 관리

- 가상환경 조회

```
(base) C:\Users\[redacted]>conda env list
# conda environments:
#
base                  *  C:\Users\[redacted]\anaconda3
```

# Anaconda 가상 환경 관리

- 새로운 가상환경 생성

```
(base) C:\Users\...> conda create -n py38 python=3.8
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 22.9.0
  latest version: 23.3.1

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\...\anaconda3\envs\py38

added / updated specs:
- python=3.8
```

## 가상환경 생성

> conda create -n [가상환경이름]

## 파이썬 버전을 지정하며 가상환경 생성하기

> conda create -n [가상환경이름] python=[0.0]

\* 가상환경 이름은 보통 프로젝트 이름이거나, 파이썬, 특정 라이브러리 버전 정보를 포함하여 지정함

ex) conda create -n project\_A

ex) conda create -n py38 python=3.8

# Anaconda 가상 환경 관리

- 가상환경 실행 & 종료

```
(base) C:\Users\>conda activate py38  
(py38) C:\Users\>
```

## 가상환경 실행

> conda activate [가상환경이름]

```
(py38) C:\Users\>conda deactivate  
(base) C:\Users\>
```

## 가상환경 종료

> conda deactivate

# Anaconda 가상 환경 관리

- 가상환경 내 패키지 목록 확인

```
(base) C:\Users\> conda list -n py38
# packages in environment at C:\Users\ \anaconda3\envs\py38:
#
# Name                          Version                      Build      Channel
ca-certificates                 2023.01.10                  haa95532_0
libffi                          3.4.4                      hd77b12b_0
openssl                         1.1.1t                      h2bbff1b_0
pip                             23.0.1                      py38haa95532_0
python                          3.8.16                      h6244533_3
setuptools                      66.0.0                      py38haa95532_0
sqlite                          3.41.2                      h2bbff1b_0
vc                              14.2                        h21ff451_1
vs2015_runtime                  14.27.29016                 h5e58377_2
wheel                           0.38.4                      py38haa95532_0
```

# Anaconda 가상 환경 관리

- 가상환경 삭제

```
(base) C:\Users\> conda remove -n py38 --all  
Remove all packages in environment C:\Users\anaconda3\envs\py38:  
  
## Package Plan ##  
  
environment location: C:\Users\anaconda3\envs\py38
```

> conda remove -n [가상환경이름] --all



에디터

# 1. Jupyter notebook

- 웹 브라우저에서 파이썬 코드를 작성하고 실행해 볼 수 있는 개발 도구
- Anaconda를 설치하면 Jupyter Notebook이 함께 설치되어 Jupyter 사용 가능

# Jupyter notebook 실행

```
(py38) C:\Users\>jupyter notebook
[I 11:04:15.323 NotebookApp] Writing notebook server cookie secret to C:\Users\AppData\Roaming\jupyter\
runtime\notebook_cookie_secret
[I 2023-05-20 11:04:18.152 LabApp] JupyterLab extension loaded from C:\Users\anaconda3\lib\site-package
s\jupyterlab
[I 2023-05-20 11:04:18.152 LabApp] JupyterLab application directory is C:\Users\anaconda3\share\jupyter
\lab
[I 11:04:18.156 NotebookApp] The port 8888 is already in use, trying another port.
[I 11:04:18.158 NotebookApp] Serving notebooks from local directory: C:\Users\Lily
[I 11:04:18.158 NotebookApp] Jupyter Notebook 6.4.0 is running at:
[I 11:04:18.158 NotebookApp] http://localhost:8889/?token=1e876d90911501c2ea55c8457c94495147b7ae18722c0cda
[I 11:04:18.158 NotebookApp] or http://127.0.0.1:8889/?token=1e876d90911501c2ea55c8457c94495147b7ae18722c0c
da
[I 11:04:18.158 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip conf
irmation).
[C 11:04:18.250 NotebookApp]
```

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

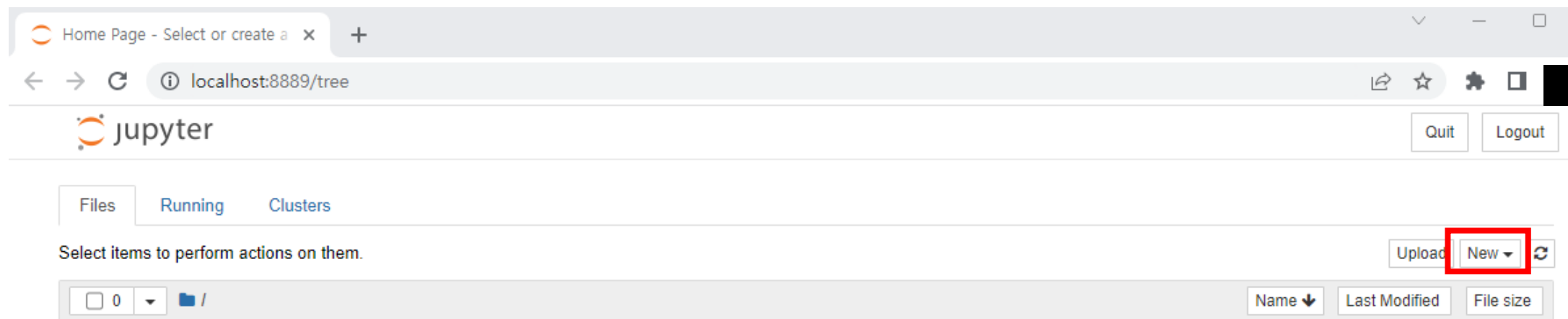
file:///C:/Users/AppData/Roaming/jupyter/runtime/nbserver-33856-open.html

Or copy and paste one of these URLs:

http://localhost:8889/?token=1e876d90911501c2ea55c8457c94495147b7ae18722c0cda

or http://127.0.0.1:8889/?token=1e876d90911501c2ea55c8457c94495147b7ae18722c0cda

# Jupyter notebook 실행



```
In [1]: print("hello")
```

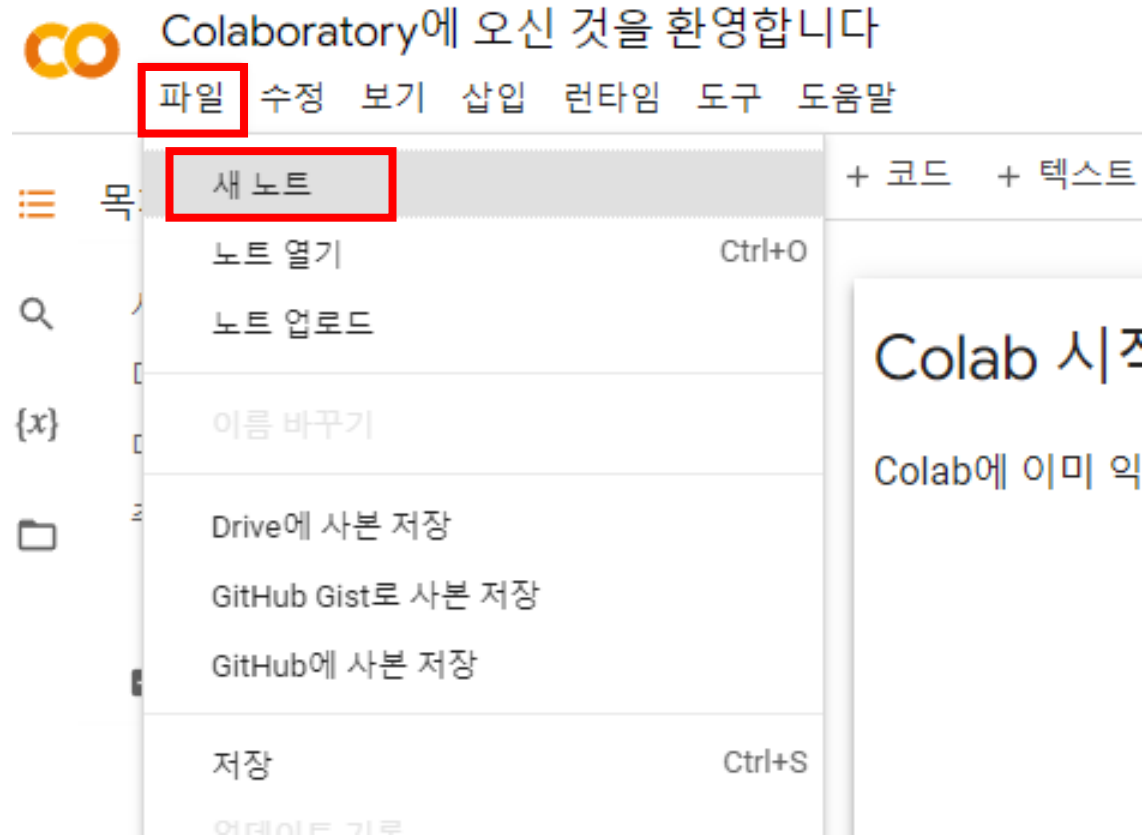
```
hello
```

## 2. Google Colab

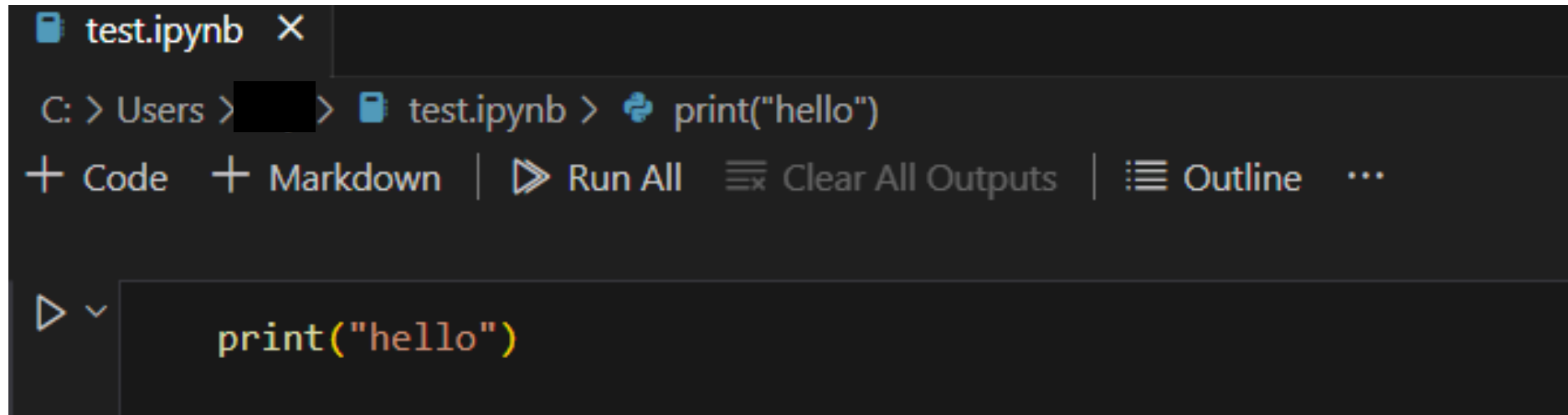
- 클라우드 기반으로 주피터 노트북 개발환경
- 브라우저에서 python을 작성하고 실행 가능하도록 해주는 에디터
- 별도의 파이썬 설치가 필요 없음
- 데이터에 분석 사용되는 Tensor Flow, Keras, matplotlib, pandas 등과 같은 패키지들이 기본적으로 설치되어 있음
- GPU를 무료로 사용할 수 있음

# Google Colab

- <https://colab.research.google.com/>



### 3. Visual Studio Code



```
test.ipynb X
```

C: > Users > <redacted> > test.ipynb > print("hello")

+ Code + Markdown | ▶ Run All ≡ Clear All Outputs | ≡ Outline ...

▶ print("hello")

.ipynb



# .ipynb

- IPython notebooks의 약자
- 과학자들이 파이썬 언어와 데이터로 작업 할 수 있도록 도와주는 인터랙티브 한 계산 환경인 Jupyter Notebook에서 사용되는 노트 파일
- 계산, 수학, 이미지 및 설명 텍스트의 입력과 출력을 포함

# 단축키 (jupyter notebook & vs code)

- **Ctrl + Enter** : 실행하기
- **Shift+ Enter** : 실행 후 다음 줄로 넘어가기
- **Esc → dd** : 셀 삭제하기

# 실습. ipynb 실행시켜보기

- 그 동안 배웠던 파이썬 코드 jupyter notebook 환경(.ipynb파일)에서 실행시켜보기