

# Qiskit 개발환경 세팅

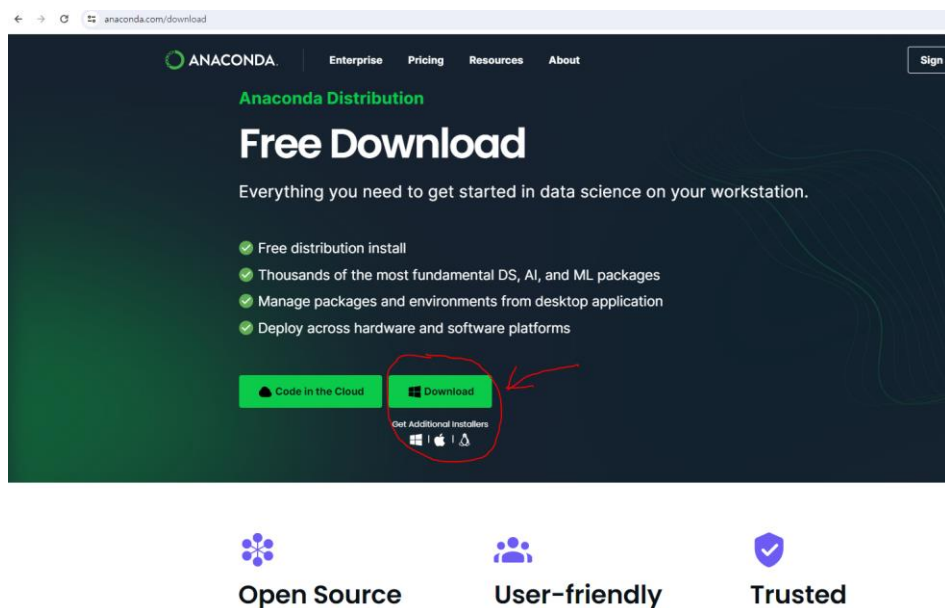
## 1. 가상 환경이 필요한 이유

가상 환경은 **독립적인 작업 환경**을 만들기 위해 사용합니다. 프로젝트를 진행하다 보면 여러 라이브러리, 패키지를 다운로드하여 사용합니다. 그러다 보면 각 라이브러리들끼리 충돌을 일으키는 문제를 발생시키는 경우가 발생합니다. 또는 A라는 라이브러리가 B라는 라이브러리의 특정 버전과 호환하는 경우 최신 버전을 사용하지 말고 이전 버전을 사용해야 하는 경우가 있습니다. 라이브러리들끼리 의존성이 많아지다보면 충돌이 일어나 잘못하면 전부 삭제하고 다시 처음부터 개발 환경을 세팅해야 하는 경우가 발생할 수 있습니다.

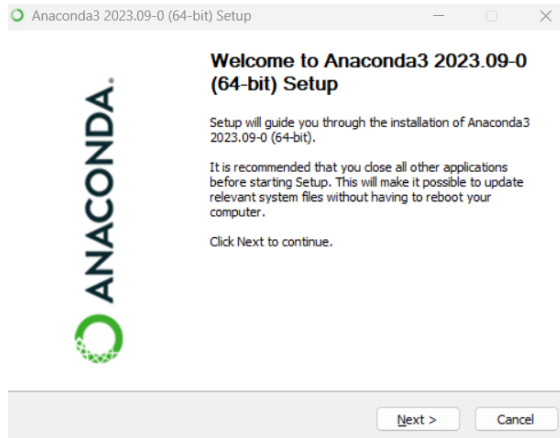
이를 방지하기 위해서 프로젝트 단위로 독립적인 가상환경을 구성하여 필요한 라이브러리를 설치해서 작업을 진행하면 작업을 할 때 편합니다. 가상환경은 활성화 및 비활성화가 가능하고 패키지 설치를 잘못 진행하였을 때 가상환경을 손쉽게 삭제할 수 있기 때문에 많이 사용합니다. 파이썬 가상환경을 만드는 방법은 여러가지가 있습니다. 그 중에서 anaconda는 의존성 체크를 해주기 때문에 많은 사람들이 사용합니다.

## 2. Anaconda 설치하기

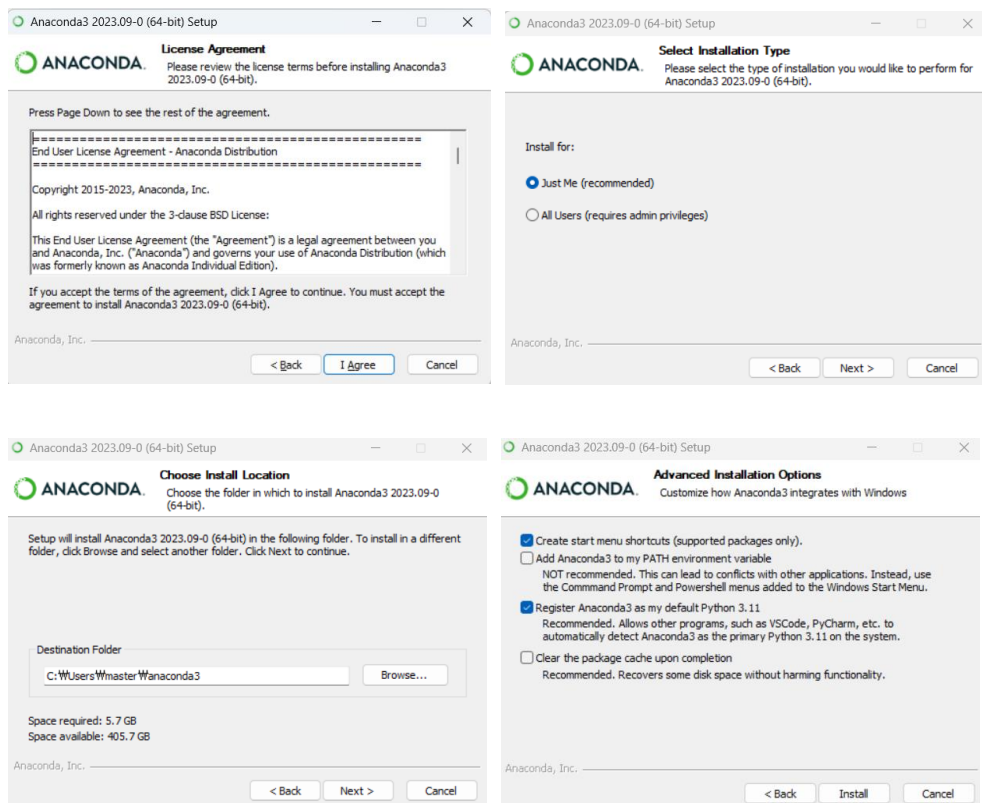
- A. <https://www.anaconda.com/download> 앞의 링크를 통해 anaconda 다운로드 웹페이지에 접속합니다.
- B. 여러분의 운영체제에 맞는 Anaconda installer를 다운로드 합니다. 요즘에는 각 운영체제에 맞춰서 추천 installer가 바로 나오는 것 같습니다.

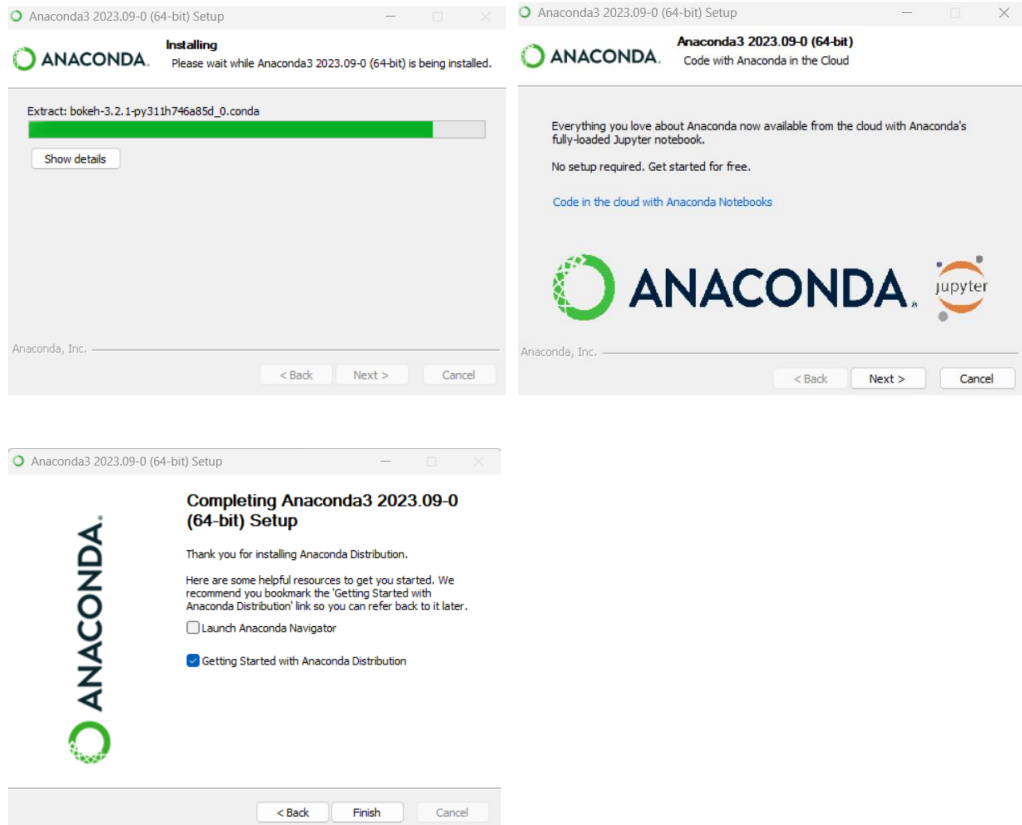


- C. Installer가 다운로드 되었다면 Installer를 실행시켜주세요. 그러면 다음과 같은 창이 나오게 될 것입니다.



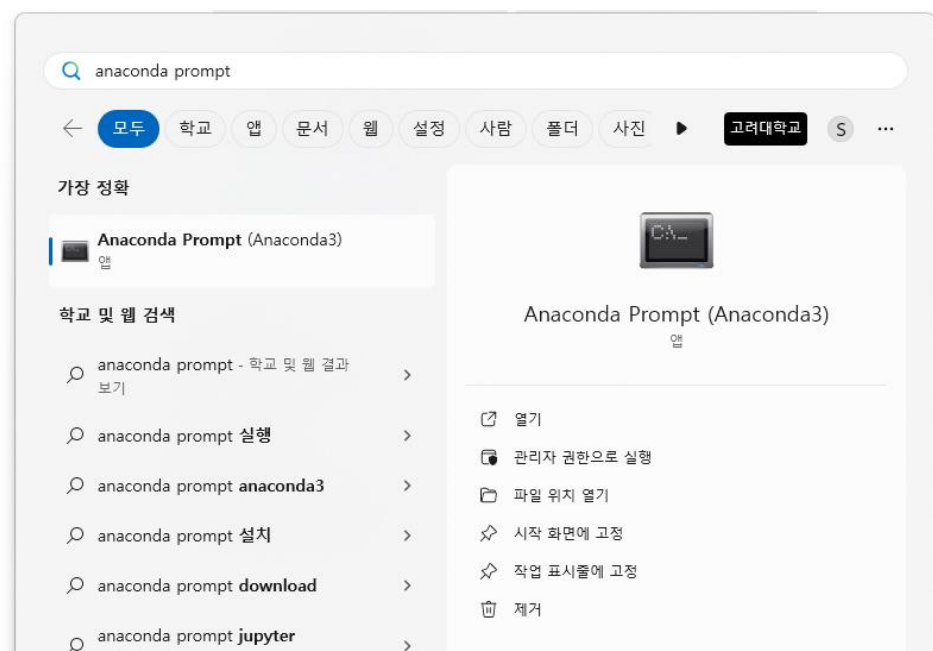
- D. 이제부터는 설정하고 싶은 사항들을 체크하면서 Next 버튼을 누르며 설치 과정을 진행하시면 됩니다. 저는 아래와 같이 선택하며 설치를 하였습니다. 아래 사진 중 3번째 사진에서는 설치 경로를 설정하는 것인데, 설치 경로에 한국어가 포함되면 설치가 안 될 수 있으니 이 점 유의 바랍니다.





### 3. Anaconda Prompt 익히기

- A. Anaconda를 다운 받으셨다면 Anaconda Prompt를 실행시켜야 합니다. 아래의 그림에 해당하는 것을 실행시키시면 됩니다.

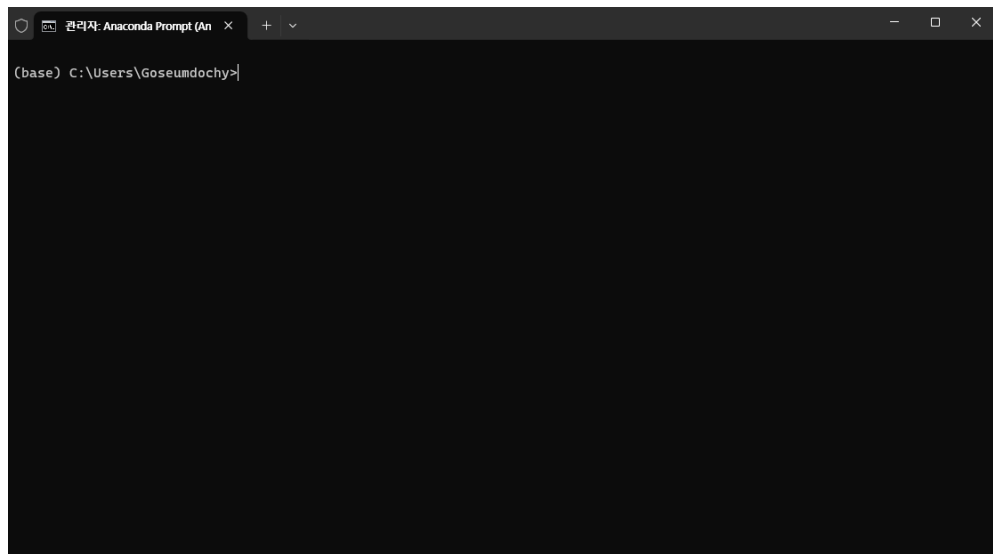


- B. 실행시키면 아래의 그림과 같은 창이 나타나게 됩니다. Anaconda Prompt에서는 커맨드를 입력하여 편리하게 개발환경을 세팅할 수 있습니다. 이렇게 텍스트 터미널을 이용하여 사용자와 컴퓨터가 상호작용하는 인터페이스를 CLI 라고 합니다.

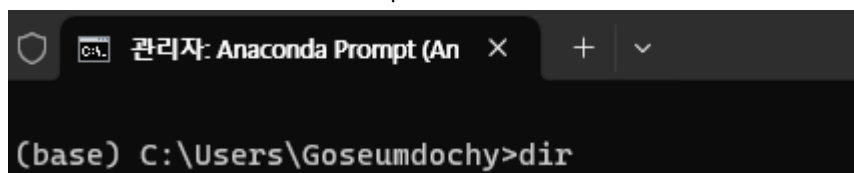
아래 그림의 가장 왼쪽 (base)는 현재 기본적인 가상환경 base가 활성화되어 있다는 뜻입니다.

그 옆의 C:\Users\Goseumdochy는 현재 **작업 디렉토리**를 의미합니다.

Goseumdochy는 저의 개인적인 user 이름이기 때문에 여러분은 다르게 나올 수 있습니다.



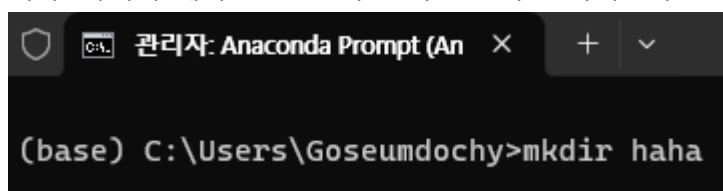
- C. 지금부터는 cd, mkdir, dir, cls와 같은 기본적인 명령어를 살펴보겠습니다. 먼저 **dir**부터 보겠습니다. Anaconda Prompt에 dir을 입력하고 Enter 키를 눌러주세요



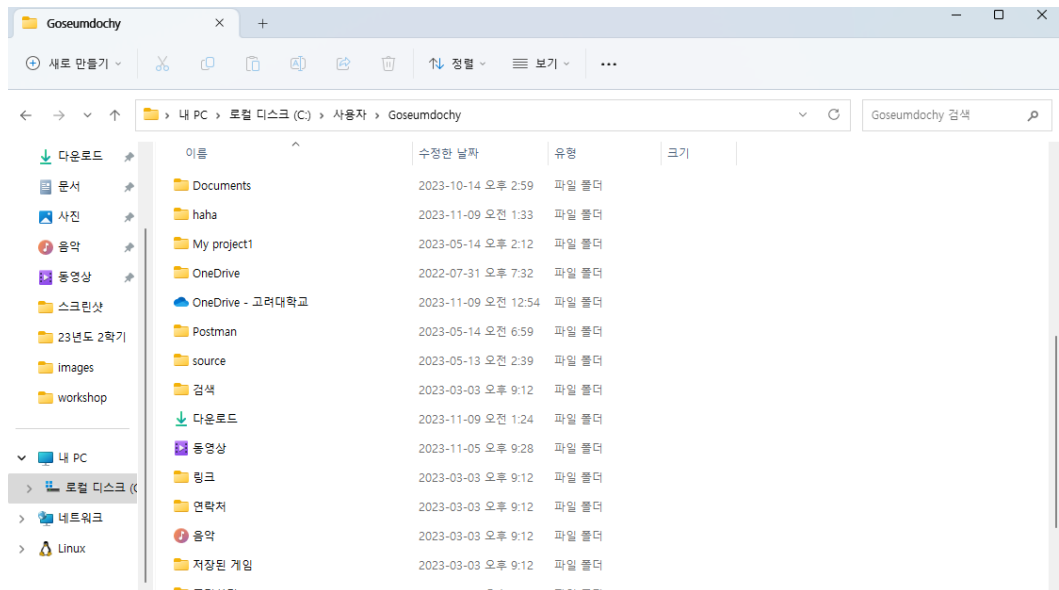
위와 같이 dir을 입력하고 Enter 키를 눌러주시면 그 밑에 현재 디렉토리 내부 폴더 및 파일 리스트가 나오게 됩니다.

- D. 다음은 새로운 디렉토리를 만드는 명령어인 **mkdir**를 보겠습니다.

**mkdir <디렉토리 이름>**을 입력하면 현재 작업 디렉토리의 하위 디렉토리가 생성됩니다. 아래의 예시는 haha 라는 이름을 가진 디렉토리를 생성한 것입니다.



디렉토리가 잘 생성되었는지는 파일 탐색기에서 보시거나 dir명령어를 이용하여 보시면 됩니다. 아래의 그림을 보시면 저 같은 경우 파일 탐색기에서 haha라는 폴더가 생긴 것을 볼 수 있습니다.

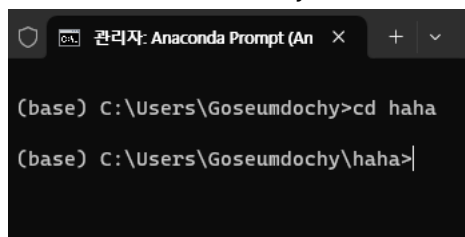


그리고 아래의 그림은 Anaconda Prompt에 dir을 입력하였을 때 haha 디렉토리가 보이는 모습입니다.

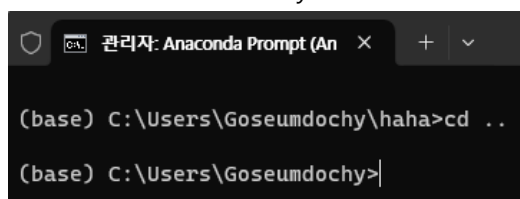


E. 다음은 작업 디렉토리를 이동하는 명령어인 cd를 보겠습니다.

cd <디렉토리>를 통해 작업 디렉토리를 이동할 수 있습니다. 방금 만든 haha라는 디렉토리로 이동해보겠습니다. 아래의 그림을 보시면 cd haha를 입력하니 C:\Users\Goseumdochy\haha로 작업 디렉토리가 변경된 것을 보실 수 있습니다.



현재 디렉토리보다 한 단계 위의 상위 디렉토리로 가기 위해서는 cd .. 명령어를 사용하시면 됩니다. 아래의 그림을 보시면 작업 디렉토리가 다시 C:\Users\Goseumdochy 가 된 것을 확인하실 수 있습니다.



- F. `cls`는 Anaconda Prompt 화면을 정리해주는 명령어입니다. 직접 Anaconda Prompt에 `cls`를 입력해보시길 바랍니다.

#### 4. 가상환경 생성하기

- A. 이제 Qiskit을 사용하기 위한 가상환경을 생성하겠습니다. 먼저 `conda create -n qiskit_test python=3.10` 을 입력해주세요. Python 3.10버전으로 작동하는 "qiskit\_test"이라는 이름을 가진 가상환경을 생성하는 명령어입니다.

```
(base) C:\Users\Goseumdochy>conda create -n qiskit_test python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

- B. 아래 그림과 같이 Proceed ([y]/n)? 이 나오면 y를 입력하고 Enter 키를 누르시길 바랍니다. 그러면 새로운 가상환경 qiskit\_test가 만들어집니다.

```
The following NEW packages will be INSTALLED:

bzip2                pkgs/main/win-64::bzip2-1.0.8-he774522_0
ca-certificates      pkgs/main/win-64::ca-certificates-2023.08.22-haa95532_0
libffi               pkgs/main/win-64::libffi-3.4.4-hd77b12b_0
openssl              pkgs/main/win-64::openssl-3.0.12-h2bbff1b_0
pip                  pkgs/main/win-64::pip-23.3-py310haa95532_0
python                pkgs/main/win-64::python-3.10.13-he1021f5_0
setuptools           pkgs/main/win-64::setuptools-68.0.0-py310haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
tk                   pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
tzdata               pkgs/main/noarch::tzdata-2023c-h04d1e81_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/win-64::wheel-0.41.2-py310haa95532_0
xz                   pkgs/main/win-64::xz-5.4.2-h8cc25b3_0
zlib                 pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0

Proceed ([y]/n)? y|
```

- C. 가상환경이 잘 만들어졌는지 확인해 보겠습니다. Anaconda Prompt에 `conda env list`를 입력하시길 바랍니다. 명령 `conda env list`는 갖고 있는 가상환경 리스트를 알려달라는 명령입니다. 그럼 아래와 같이 qiskit\_test 라는 이름을 가진 가상환경이 있는 것을 확인할 수 있습니다. 저는 이전에 다른 가상환경을 세팅해서 사용해왔기에 여러 가상환경이 나오지만, Anaconda를 처음 설치하신 분은 base와 qiskit\_test 두 개의 가상환경이 보일 것입니다.

base 오른쪽에 \* 표시는 현재 활성화된 가상환경을 나타내는 것입니다.

```
(base) C:\Users\Goseumdochy>conda env list
# conda environments:
#
base                * C:\ProgramData\Anaconda3
Qiskit              C:\ProgramData\Anaconda3\envs\Qiskit
manim-project       C:\ProgramData\Anaconda3\envs\manim-project
pennylane           C:\ProgramData\Anaconda3\envs\pennylane
qff_ku2023          C:\ProgramData\Anaconda3\envs\qff_ku2023
qff_ku2023_test     C:\ProgramData\Anaconda3\envs\qff_ku2023_test
qiskit_seminar      C:\ProgramData\Anaconda3\envs\qiskit_seminar
qiskit_test         C:\ProgramData\Anaconda3\envs\qiskit_test
```

## 5. 가상환경 활성화

- A. Anaconda Prompt에 `conda activate qiskit_test`를 입력하시길 바랍니다. 방금 생성한 가상환경을 활성화시키는 명령입니다. 아래의 그림에서는 그 이후 `conda env list`를 입력하여 \* 표시가 qiskit\_test의 오른쪽에 나타나는 것을 확인하였습니다. 작업 디렉토리 왼쪽이 (base)에서 (qiskit\_test)로 바뀐 것도 확인하실 수 있습니다.

```
(base) C:\Users\Goseumdochy>conda activate qiskit_test

(qiskit_test) C:\Users\Goseumdochy>conda env list
# conda environments:
#
base                  C:\ProgramData\Anaconda3
qiskit                C:\ProgramData\Anaconda3\envs\qiskit
manim-project         C:\ProgramData\Anaconda3\envs\manim-project
pennylane             C:\ProgramData\Anaconda3\envs\pennylane
qff_ku2023            C:\ProgramData\Anaconda3\envs\qff_ku2023
qff_ku2023_test       C:\ProgramData\Anaconda3\envs\qff_ku2023_test
qiskit_seminar        C:\ProgramData\Anaconda3\envs\qiskit_seminar
qiskit_test           * C:\ProgramData\Anaconda3\envs\qiskit_test

(qiskit_test) C:\Users\Goseumdochy>
```

## 6. 가상 환경에 라이브러리 설치

- A. 이제 qiskit\_test 가상환경에 Qiskit 개발을 위한 패키지들을 설치하겠습니다. 가상환경 qiskit\_test가 활성화된 상태에서 `conda install jupyter notebook`을 입력하세요. Proceed ([y]/n)? 이 나오면 y를 입력하고 Enter 키를 누르세요. 잠시 기다리면 다운로드가 될 것입니다.

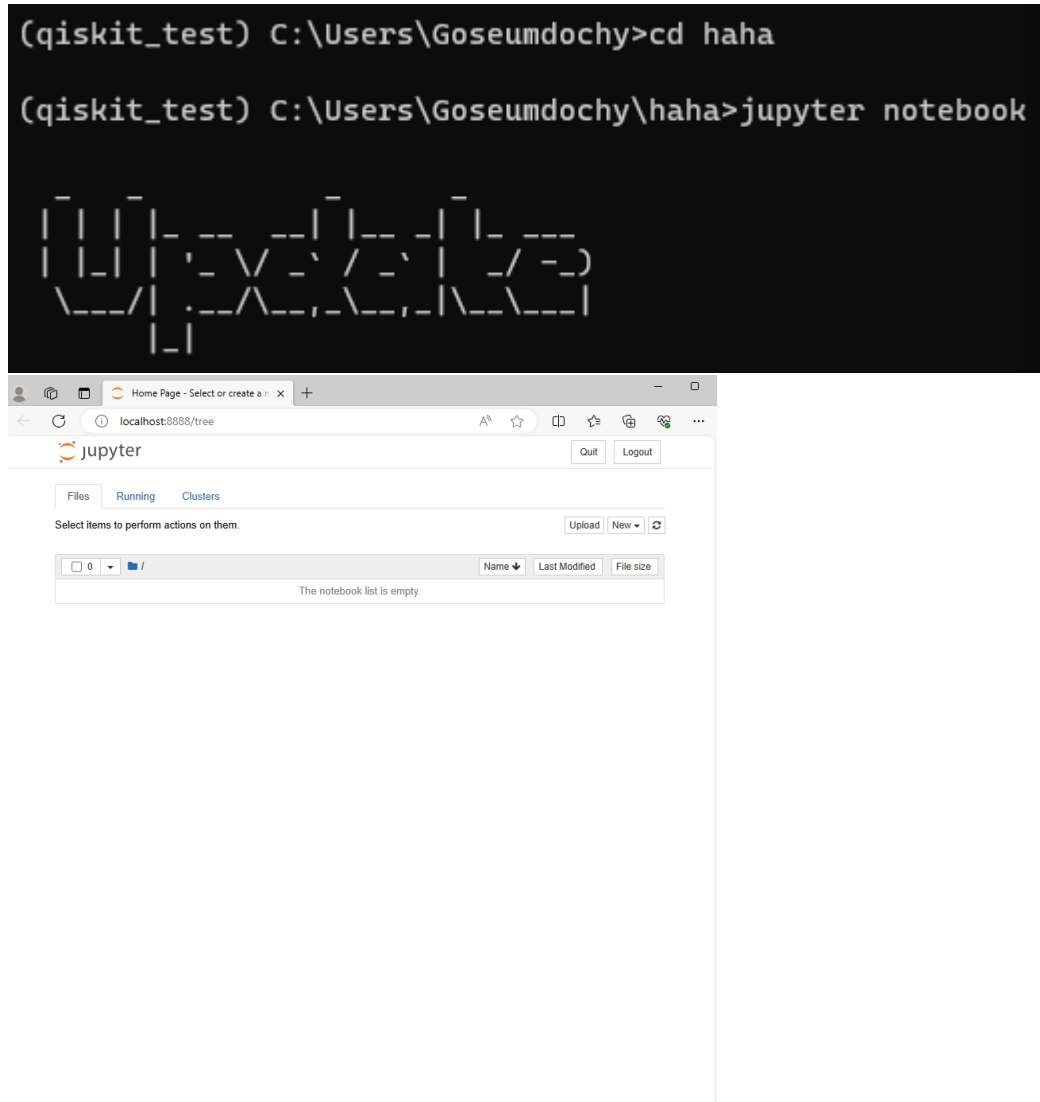
```
(qiskit_test) C:\Users\Goseumdochy>conda install jupyter notebook
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

- B. 그 다음 명령 `pip install qiskit[all]`을 입력하세요.
- C. 그 다음 명령 `pip install qiskit qiskit_ibm_provider`을 입력하세요.
- D. 그 다음 명령 `pip install qiskit-ibm-runtime qiskit-aer`을 입력하세요.

## 7. 주피터 노트북 실행 및 테스트

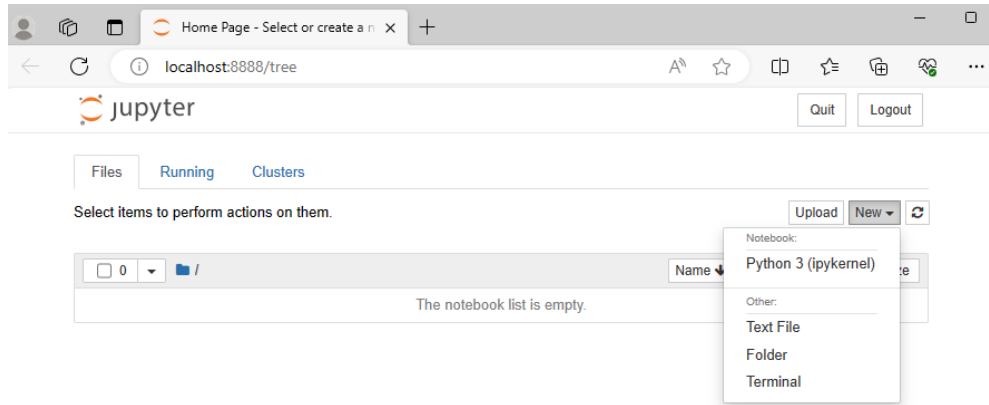
- A. 앞에서 가상환경 세팅이 끝났습니다! 이제 주피터 노트북을 실행시켜봅시다. 그 전에 저는 `cd haha`를 통해 `haha` 디렉토리로 이동하겠습니다.

Anaconda Prompt에 `jupyter notebook`을 입력하세요. 잠시 기다리면 웹페이지가 나타나게 될 것입니다.

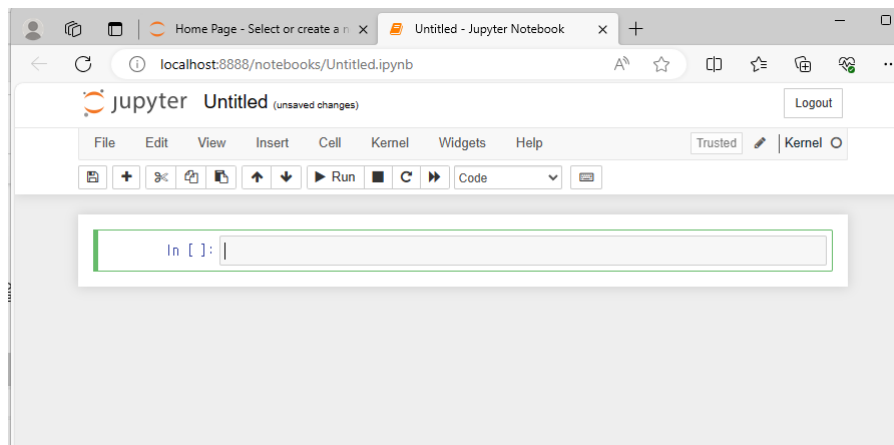




- B. 이 웹페이지 우측 상단에 New버튼을 클릭하고 Python 3 (ipykernel) 항목을 클릭합니다.

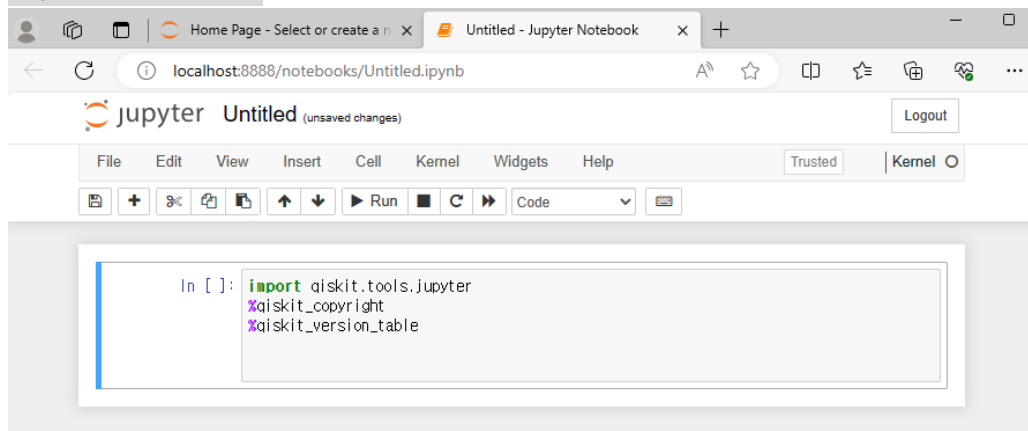


- C. 아래와 같은 창이 나올 것입니다.



- D. 코드셀에 아래 그림과 같이 입력합니다.

```
import qiskit.tools.jupyter
%qiskit_copyright
%qiskit_version_table
```



- E. 코드셀을 클릭한 상태에서 Shift+Enter 키를 누르시길 바랍니다. Shift+Enter는 현재 선택된 코드셀을 실행시키는 주피터 노트북의 단축키입니다. 이 작업을 수행했을 때 아래 그림과 같은 화면이 나온다면 Qiskit 개발 환경 세팅 성공입니다!

```
In [1]: import qiskit.tools.jupyter
        %qiskit_copyright
        %qiskit_version_table
```

#### This code is a part of Qiskit

© Copyright IBM 2017, 2023.

This code is licensed under the Apache License, Version 2.0. You may obtain a copy of this license in the LICENSE.txt file in the root directory of this source tree or at <http://www.apache.org/licenses/LICENSE-2.0>.

Any modifications or derivative works of this code must retain this copyright notice, and modified files need to carry a notice indicating that they have been altered from the originals.

#### Version Information

Software	Version
qiskit	0.45.0
System information	
Python version	3.10.13
Python compiler	MSC v.1916 64 bit (AMD64)
Python build	main, Sep 11 2023 13:24:38
OS	Windows
CPUs	8
Memory (Gb)	7.247257232666016
Thu Nov 09 02:36:05 2023 대한민국 표준시	