

# Setting Up the Qiskit Development Environment

## 1. Why Virtual Environments Are Necessary

Virtual environments are used to create isolated workspaces. When working on a project, you often need to download and use multiple libraries and packages. However, conflicts can arise between these libraries. For example, although library A is compatible with a specific version of library B, you may need to use a different version of library B for some project instead of the compatible version. As the number of dependencies between libraries increases, these conflicts can become more frequent, and in the worst case, you might have to delete everything and set up the development environment from scratch.

To avoid these issues, it's helpful to create a separate virtual environment for each project, where only the necessary libraries are installed. Virtual environments can be easily activated or deactivated, and if you encounter problems (e.g., installing a package incorrectly), you can simply delete the virtual environment and start over. This makes virtual environments a valuable tool for developers.

There are several ways to create virtual environments in Python, and one popular option is using Anaconda, which simplifies dependency management and is widely used for this reason.

## 2. Installing Anaconda

A. Go to the Anaconda download page through the link:

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

B. Download the proper Anaconda installer that matches your operating system.

Nowadays, the recommended installer for your OS is usually shown automatically.

## Download Now

For installation assistance, refer to [Troubleshooting](#).

Download Anaconda Distribution or [Miniconda](#) by choosing the proper installer for your machine. Learn the difference from our [Documentation](#).



### Anaconda Installers

[Download](#)

#### Windows

**Python 3.12**

- 64-Bit Graphical Installer (912.3M)

#### Mac

**Python 3.12**

- 64-Bit (Apple silicon) Graphical Installer (704.7M)
- 64-Bit (Apple silicon) Command Line Installer (707.3M)
- 64-Bit (Intel chip) Graphical Installer (734.7M)

#### Linux

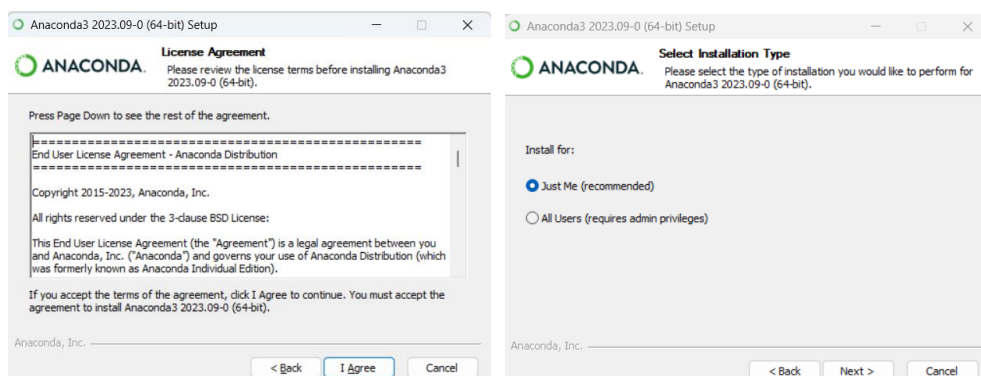
**Python 3.12**

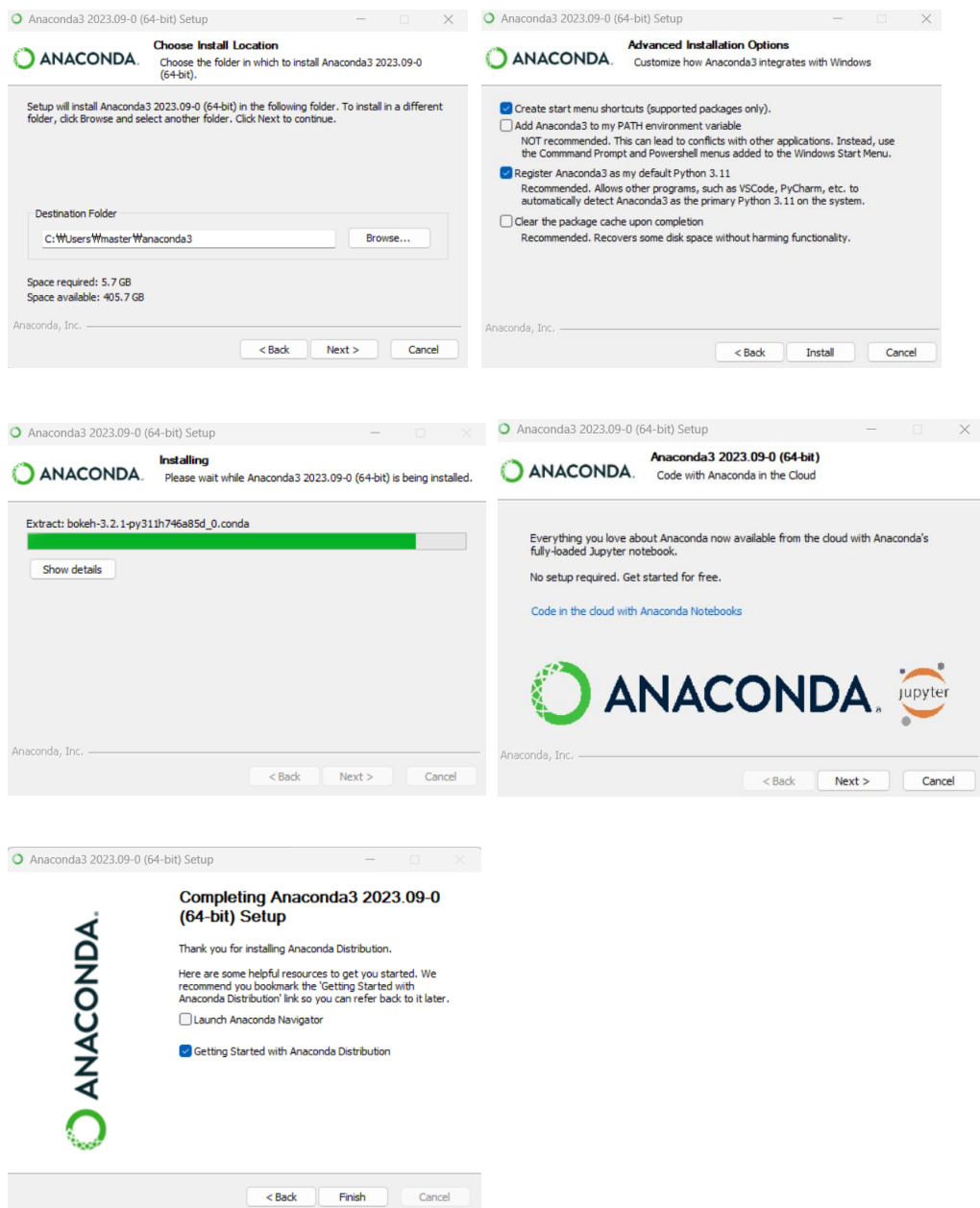
- 64-Bit (x86) Installer (1007.9M)
- 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)
- 64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)

- C. Once the installer has been downloaded, run the installer. A window like the one below will appear.



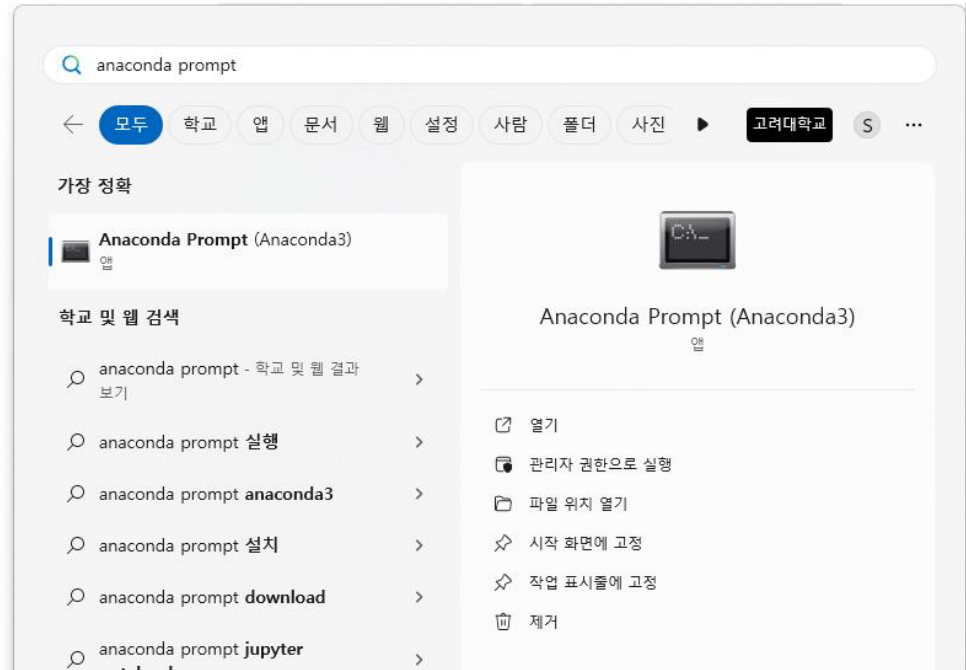
- D. From here, proceed through the installation process by checking the options you wish to configure and click "Next". I selected the options as shown below. In the third image, you'll be asked to choose the installation path. Please be careful not to include Korean characters in the installation path, as this may cause the installation to fail.



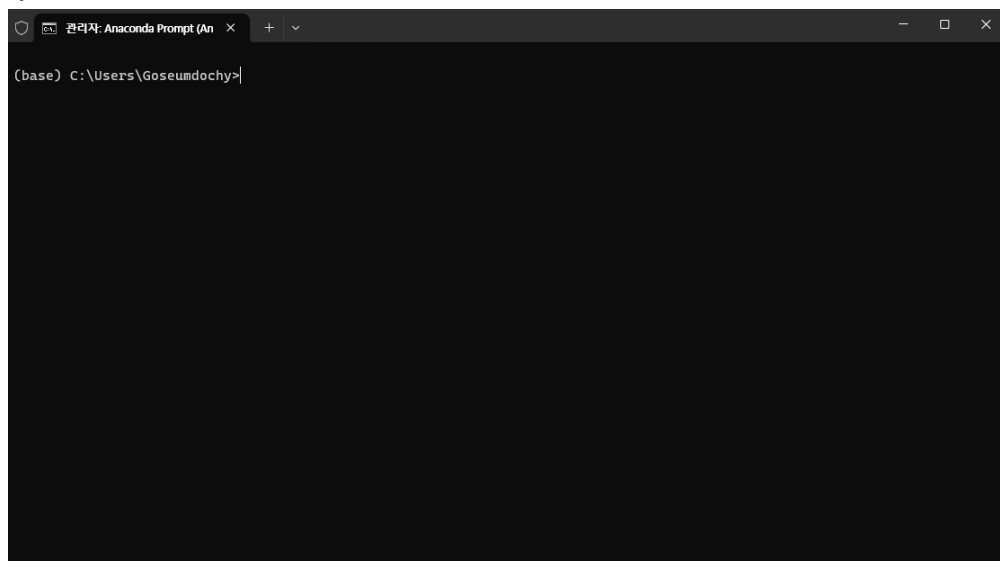


### 3. Getting Familiar with Anaconda Prompt

- A. After installing Anaconda, you need to launch the Anaconda Prompt. Run the program shown in the image below.

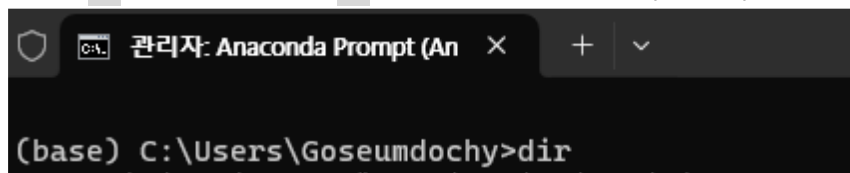


- B. When you launch it, a window like the one below will appear. The Anaconda Prompt allows you to conveniently set up your development environment by entering commands. This interface, where you interact with the computer through text commands, is called the Command Line Interface (CLI). The "(base)" on the far left indicates that the default virtual environment "base" is activated. The "C:\Users\Goseumdochy" part shows the current working directory. Since "Goseumdochy" is my personal username, it may look different on your system.



- C. Now let's go over some basic commands like `cd`, `mkdir`, `dir`, and `cls`. First, let's look

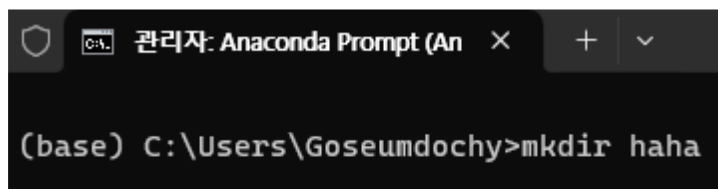
at the `dir` command. Enter `dir` in the Anaconda Prompt and press Enter.



```
(base) C:\Users\Goseumdochy>dir
```

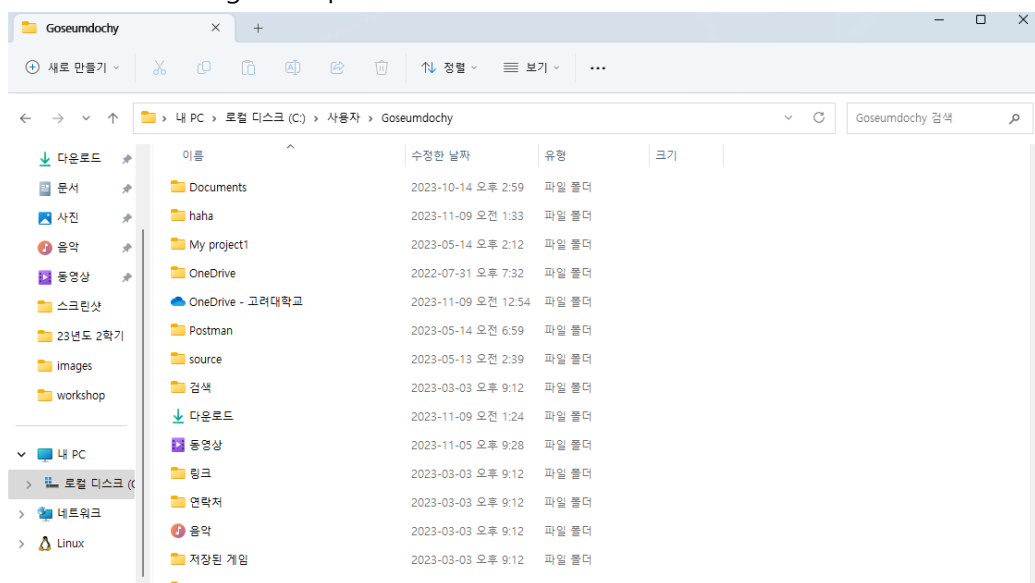
This will show the list of folders and files in the current directory.

- D. Next, let's look at the `mkdir` command, which is used to create a new directory. If you type `mkdir <directory_name>`, a subdirectory will be created in the current working directory. For example, entering `mkdir haha` will create a directory named "haha."



```
(base) C:\Users\Goseumdochy>mkdir haha
```

To confirm that the directory was created, you can either check in the file explorer or use the `dir` command again. In the image below, you can see that I created a "haha" folder using File Explorer.



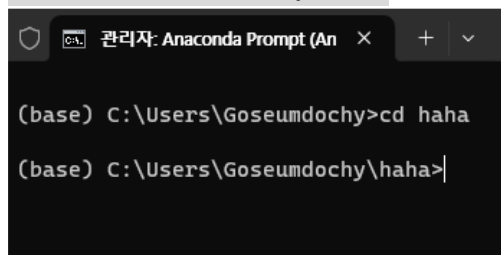
And the the directory is visible when I enter `dir` in the Anaconda Prompt.



```
2023-11-09 오전 01:33 <DIR>      haha
```

- E. Next, let's look at the `cd` command, which is used to change the working directory. By entering `cd <directory_name>`, you can navigate to a new directory. I will navigate to the "haha" directory that I just created by typing `cd haha`. In the image below, you can see that the working directory has changed to

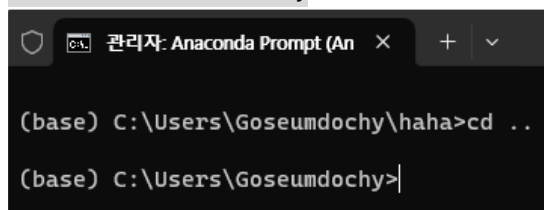
C:\Users\Goseumdochy\haha.

A screenshot of an Anaconda Prompt window. The title bar shows '관리자: Anaconda Prompt (An' with a close button. The prompt shows the current directory as C:\Users\Goseumdochy\haha. The user has entered 'cd haha' and the prompt has moved to C:\Users\Goseumdochy\haha>.

```
(base) C:\Users\Goseumdochy>cd haha
(base) C:\Users\Goseumdochy\haha>
```

To move up one level to the parent directory, use the command `cd ..`. In the image below, you can see that the directory has changed back to

C:\Users\Goseumdochy.

A screenshot of an Anaconda Prompt window. The title bar shows '관리자: Anaconda Prompt (An' with a close button. The prompt shows the current directory as C:\Users\Goseumdochy\haha. The user has entered 'cd ..' and the prompt has moved back to C:\Users\Goseumdochy>.

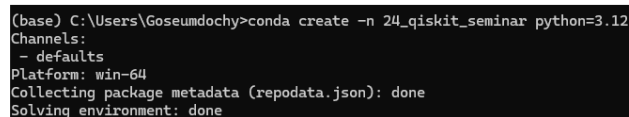
```
(base) C:\Users\Goseumdochy\haha>cd ..
(base) C:\Users\Goseumdochy>
```

- F. The `cls` command clears the screen in the Anaconda Prompt. Try typing `cls` in the Anaconda Prompt to see this in action.

#### 4. Creating a Virtual Environment

- A. Now, let's create a virtual environment for using Qiskit. First, enter the following command: `conda create -n 24_qiskit_seminar python=3.12`

This command creates a virtual environment named "24\_qiskit\_seminar" that uses Python 3.12.

A screenshot of an Anaconda Prompt window showing the output of the 'conda create' command. The output indicates that the environment was successfully created with Python 3.12.

```
(base) C:\Users\Goseumdochy>conda create -n 24_qiskit_seminar python=3.12
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

- B. When you see the prompt `Proceed ([y]/n)?`, type `y` and press Enter. This will proceed with creating the virtual environment "24\_qiskit\_seminar".

The following NEW packages will be INSTALLED:

bzip2	pkgs/main/win-64::bzip2-1.0.8-h2bbff1b_6
ca-certificates	pkgs/main/win-64::ca-certificates-2024.9.24-haa95532_0
expat	pkgs/main/win-64::expat-2.6.3-h5da7b33_0
libffi	pkgs/main/win-64::libffi-3.4.4-hd77b12b_1
openssl	pkgs/main/win-64::openssl-3.0.15-h827c3e9_0
pip	pkgs/main/win-64::pip-24.2-py312haa95532_0
python	pkgs/main/win-64::python-3.12.7-h14ffc60_0
setuptools	pkgs/main/win-64::setuptools-75.1.0-py312haa95532_0
sqlite	pkgs/main/win-64::sqlite-3.45.3-h2bbff1b_0
tk	pkgs/main/win-64::tk-8.6.14-h0416ee5_0
tzdata	pkgs/main/noarch::tzdata-2024b-h04d1e81_0
vc	pkgs/main/win-64::vc-14.40-h2eaa2aa_1
vs2015_runtime	pkgs/main/win-64::vs2015_runtime-14.40.33807-h98bb1dd_1
wheel	pkgs/main/win-64::wheel-0.44.0-py312haa95532_0
xz	pkgs/main/win-64::xz-5.4.6-h8cc25b3_1
zlib	pkgs/main/win-64::zlib-1.2.13-h8cc25b3_1

Proceed ([y]/n)?

- C. To check if the virtual environment was created successfully, enter the command:

```
conda env list
```

This command will list all the virtual environments you have. You should see "24\_qiskit\_seminar" listed. If you have previously set up other environments, you may see more than one, but if you installed Anaconda now, you should see two environments: "base" and "24\_qiskit\_seminar." The asterisk (\*) next to "base" indicates that it is the currently active environment.

```
(base) C:\Users\Goseumdochy>conda env list
# conda environments:
#
base                * C:\ProgramData\Anaconda3
24_qiskit_seminar   C:\ProgramData\Anaconda3\envs\24_qiskit_seminar
QHackathon2024      C:\ProgramData\Anaconda3\envs\QHackathon2024
Qiskit              C:\ProgramData\Anaconda3\envs\Qiskit
QuantumHackathon_2024 C:\ProgramData\Anaconda3\envs\QuantumHackathon_2024
Quantumhackathon_2024 C:\ProgramData\Anaconda3\envs\Quantumhackathon_2024
bqit2024            C:\ProgramData\Anaconda3\envs\bqit2024
manim-project       C:\ProgramData\Anaconda3\envs\manim-project
old_qiskit          C:\ProgramData\Anaconda3\envs\old_qiskit
pennylane           C:\ProgramData\Anaconda3\envs\pennylane
qcsg                C:\ProgramData\Anaconda3\envs\qcsg
qff_ku2023          C:\ProgramData\Anaconda3\envs\qff_ku2023
qff_ku2023_test     C:\ProgramData\Anaconda3\envs\qff_ku2023_test
qff_ku2024_server_test C:\ProgramData\Anaconda3\envs\qff_ku2024_server_test
qff_ku2024_test     C:\ProgramData\Anaconda3\envs\qff_ku2024_test
qffatku2023         C:\ProgramData\Anaconda3\envs\qffatku2023
qgss2024            C:\ProgramData\Anaconda3\envs\qgss2024
qiskit_seminar      C:\ProgramData\Anaconda3\envs\qiskit_seminar
qiskit_test         C:\ProgramData\Anaconda3\envs\qiskit_test
qpong              C:\ProgramData\Anaconda3\envs\qpong
test                C:\ProgramData\Anaconda3\envs\test
```

## 5. Activating the Virtual Environment

- A. To activate the virtual environment you just created, enter the command:

```
conda activate 24_qiskit_seminar
```

After running this command, you should see that the prompt has changed. The

active environment will now be show as "(24\_qiskit\_seminar)" instead of "(base)," and you can confirm this by running `conda env list`.

```
(base) C:\Users\Goseumdochy>conda activate 24_qiskit_seminar

(24_qiskit_seminar) C:\Users\Goseumdochy>
(24_qiskit_seminar) C:\Users\Goseumdochy>conda env list
# conda environments:
#
base                        C:\ProgramData\Anaconda3
24_qiskit_seminar          * C:\ProgramData\Anaconda3\envs\24_qiskit_seminar
QHackathon2024             C:\ProgramData\Anaconda3\envs\QHackathon2024
```

## 6. Installing Libraries in the Virtual Environment

- A. Now that the "24\_qiskit\_seminar" environment is active, let's install the packages required for Qiskit development. Enter the following command:

```
pip install qiskit[visualization] qiskit_aer qiskit_ibm_runtime
```

```
(24_qiskit_seminar) C:\Users\Goseumdochy>pip install qiskit[visualization] qiskit_aer qiskit_ibm_runtime
Collecting qiskit_aer
```

- B. Next, install additional packages by running the following command:

```
pip install matplotlib pylatexenc chardet seaborn
```

- C. Now, install Jupyter Lab by running:

```
conda install jupyterlab
```

When prompted with `Proceed ([y]/n)?`, type `y` and press Enter. Wait a moment for the download to complete.

## 7. Running and Testing Jupyter Notebook

- A. The virtual environment setup is complete! Now, let's launch Jupyter Notebook. Before doing that, I go to the "haha" directory by entering `cd haha`. Next, enter `jupyter lab` in the Anaconda Prompt. After a short wait, the Jupyter Lab web interface should appear in your browser.

```
(24_qiskit_seminar) C:\Users\Goseumdochy>cd haha
(24_qiskit_seminar) C:\Users\Goseumdochy\haha>jupyter lab
```



