Getting started

With the growing interest in quantum computing, there are an increasing number of development libraries and tools for the field. There are development environments and simulators in all the major languages including Python, C/C++, Java, and others. Many of the leading QC research centers have focused on Python as the language of choice for building quantum circuits. One of the reasons for choosing Python is that it is a flexible, high-level language that allows programmers to focus on the problem being solved without worrying about too many formal details. For example, Python is dynamically typed (meaning variable types do not have to be declared by the programmer) and is an interpreted language (meaning it does not have to be pre-compiled into a binary executable). For these reasons and others, Python has a relatively easy learning curve for new users and has already seen strong support from the QCcommunity.

Preparing Your Environment

Before you can start working on your quantum programs, you must have a Python environment to execute your code. The examples in this course can be run both on your local machine by using the Qiskit developer environment provided by IBM Quantum and in an online environment on IBM Quantum Experience.

We will take a look at both environments, get you a login account on IBM Quantum Experience, and install a local version of Qiskit. We will also discuss the fast-moving environment that is open source Qiskit, and how to keep your local environment up to date.

IBM Quantum recommends using the Anaconda distribution of Python (https://www.anaconda.com/download), and to use virtual environments to keep your Qiskit installation isolated from your usual Python environment.

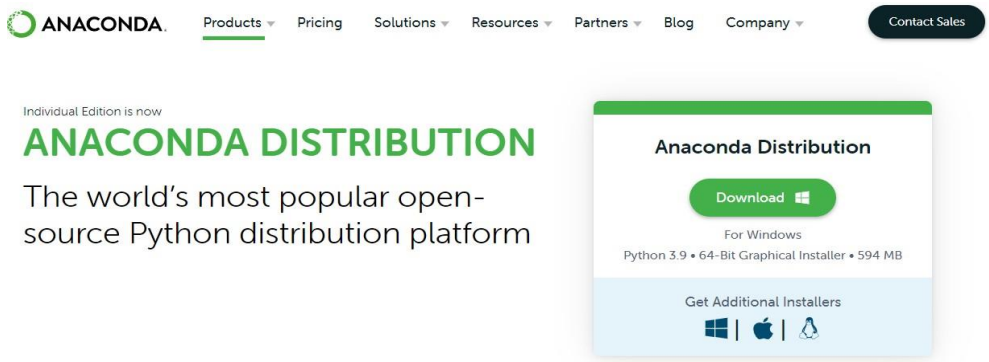
New to virtual environments?
Virtual environments provide isolated Python environments that you can modify separately from each other. For example, you can create an isolated environment for your Qiskit installation. You will then install Qiskit only in that environment, and not touch the Python framework in the base environment which will then contain an untarnished version of Python.

You will need to install Jupyter Notebook to read and run the programs on your local system. We recommend creating and using a separate virtual environment (named "QCQML").

Installation of Jupyter Notebook:

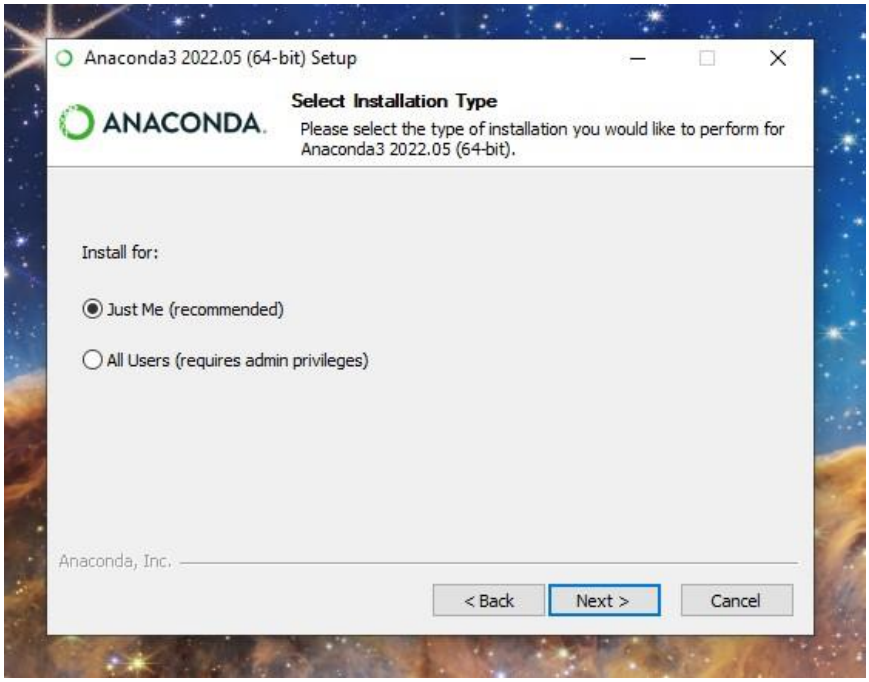
We recommend using Anaconda to install Jupyter Notebook. Installing Anaconda

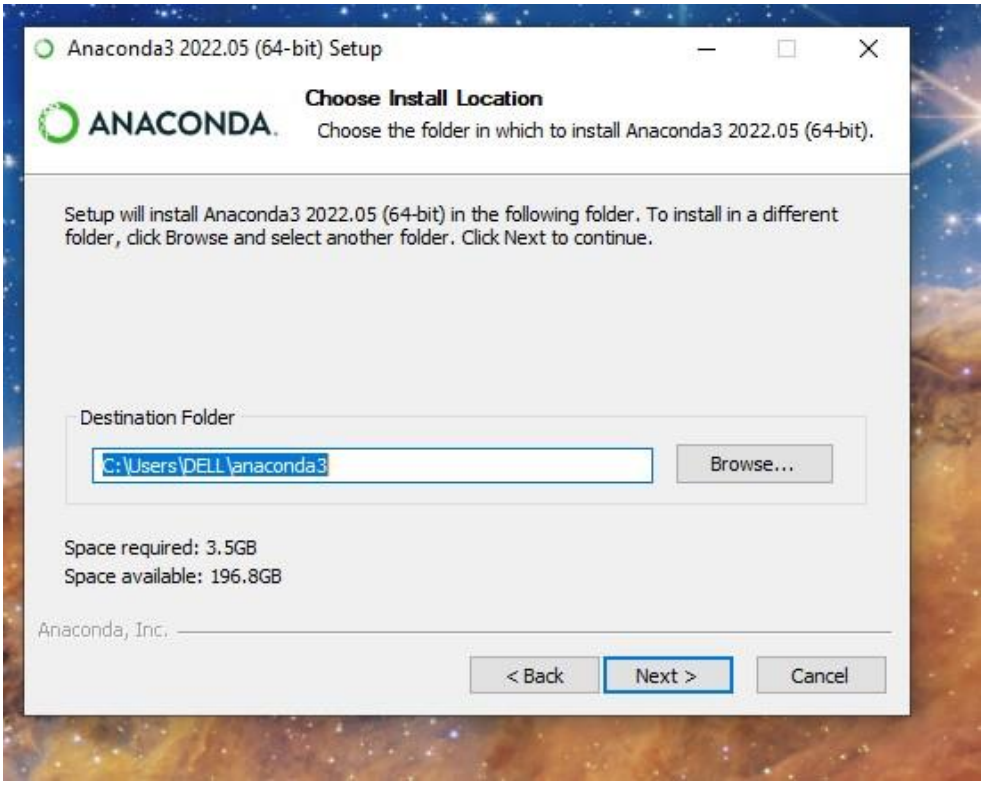
1. **Anaconda is available for multiple platforms. Anaconda Installation**

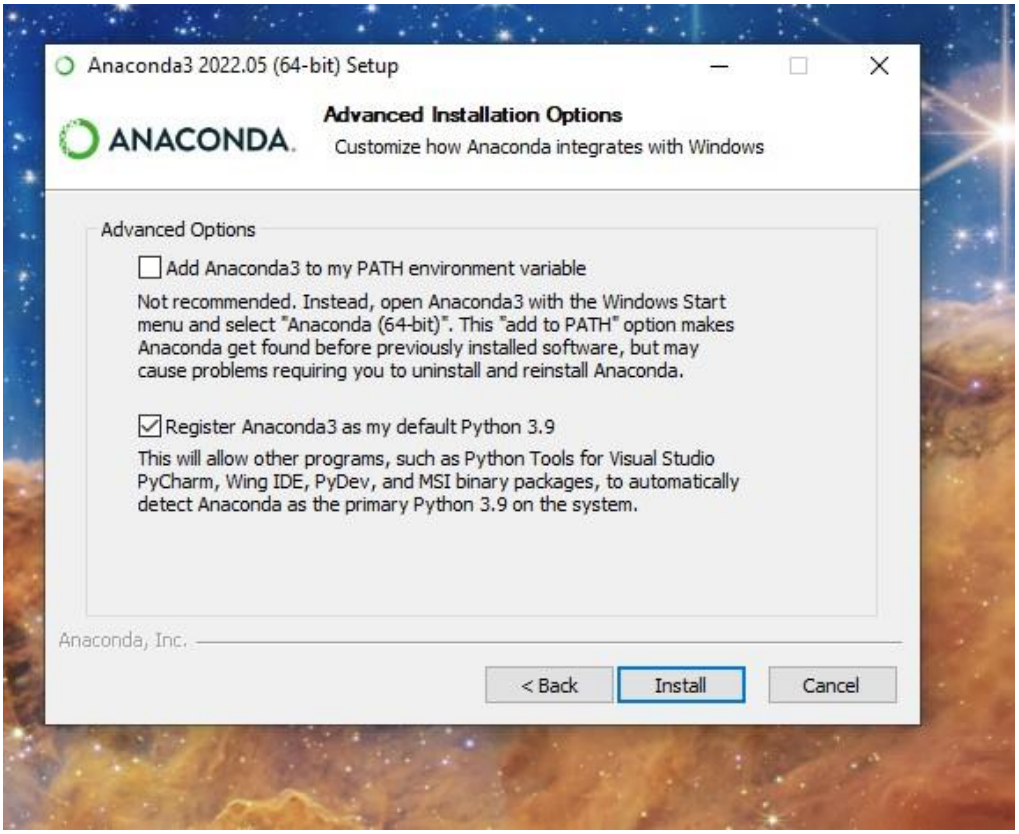


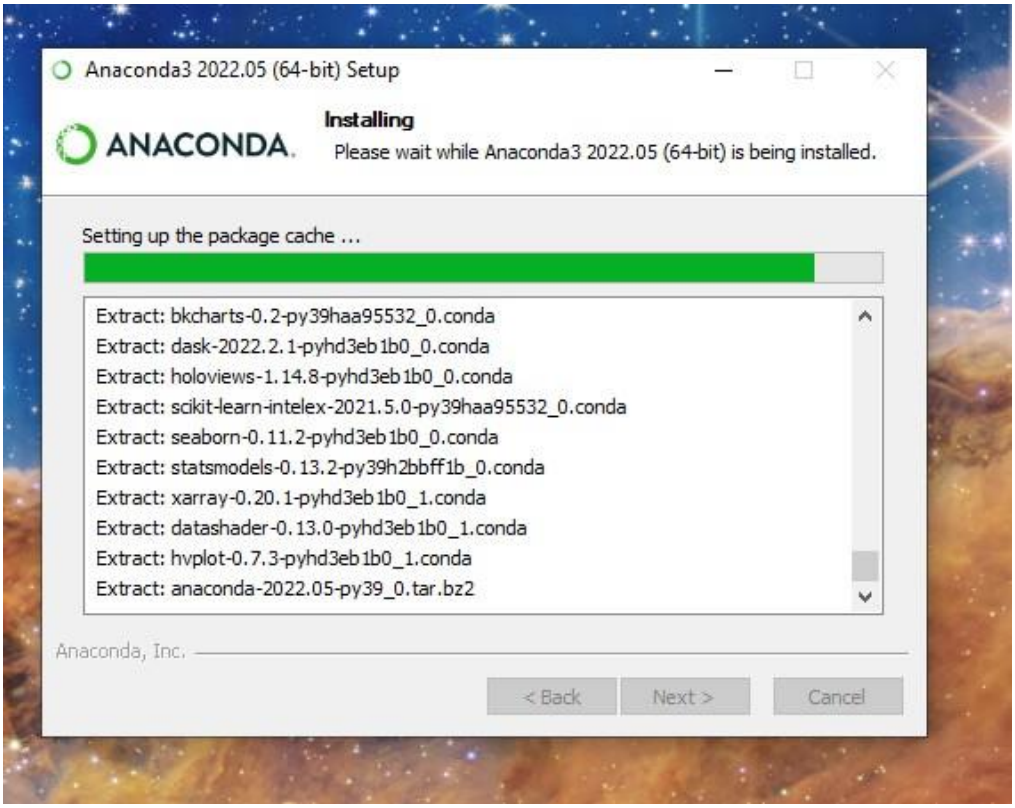
The official installation guide is https://docs.anaconda.com/anaconda/

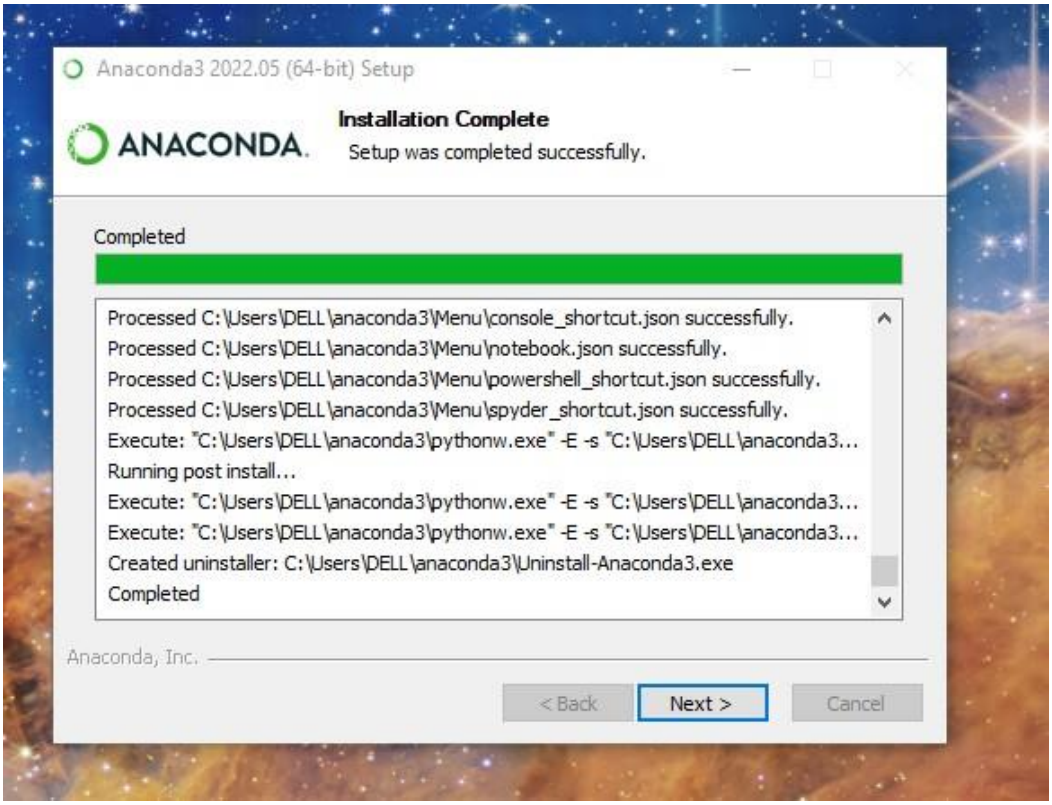
2. **Installation on Windows:**


Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Select Installation Type
Please select the type of installation you would like to perform for Anaconda3 2022.05 (64-bit).
Install for:
Just Me (recommended)
All Users (requires admin privileges)
Anaconda, Inc.
< Back
Next >
Cancel

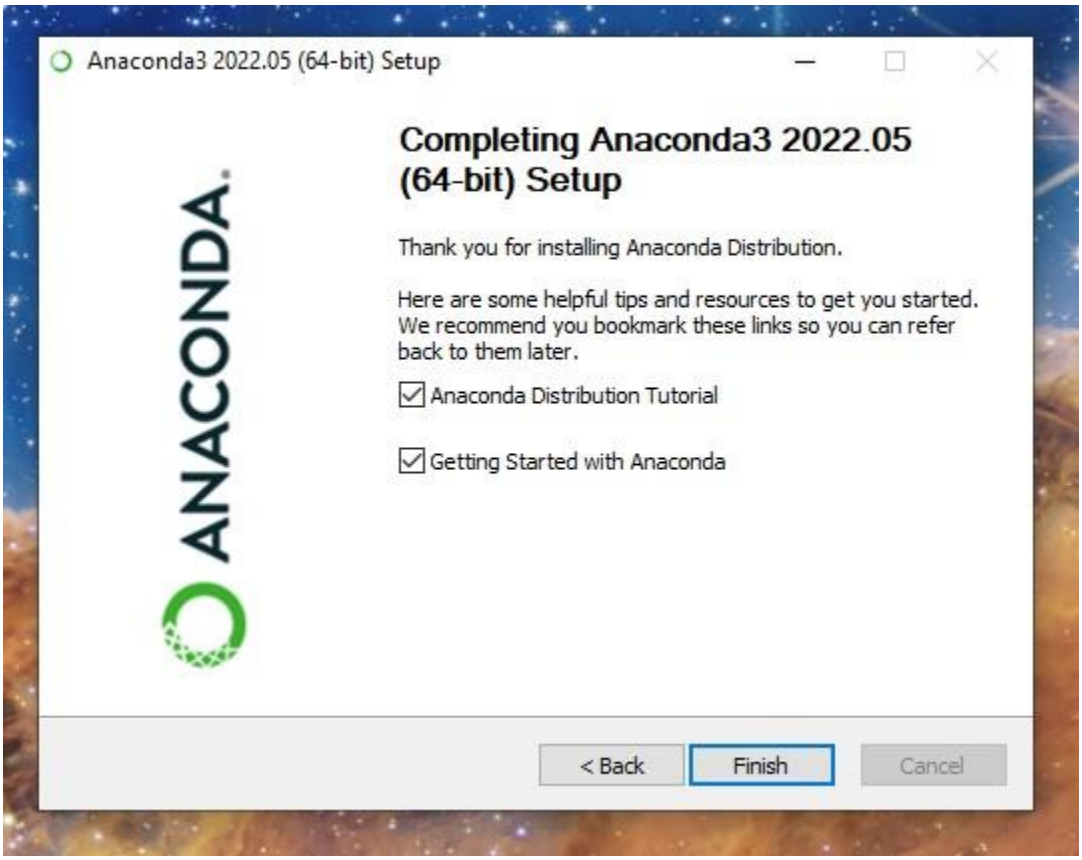

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Choose Install Location
Choose the folder in which to install Anaconda3 2022.05 (64-bit).
Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Users\DELL\anaconda3
Browse...
Space required: 3.5GB
Space available: 196.8GB
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Advanced Installation Options
Customize how Anaconda integrates with Windows
Advanced Options
Add Anaconda3 to my PATH environment variable
Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.
Register Anaconda3 as my default Python 3.9
This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.
Anaconda, Inc.
< Back
Install
Cancel


Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Installing
Please wait while Anaconda3 2022.05 (64-bit) is being installed.
Setting up the package cache ...
Extract: bkcharts-0.2-py39haa95532_0.conda
Extract: dask-2022.2.1-pyhd3eb1b0_0.conda
Extract: holoviews-1.14.8-pyhd3eb1b0_0.conda
Extract: scikit-learn-intelex-2021.5.0-py39haa95532_0.conda
Extract: seaborn-0.11.2-pyhd3eb1b0_0.conda
Extract: statsmodels-0.13.2-py39h2bbff1b_0.conda
Extract: xarray-0.20.1-pyhd3eb1b0_1.conda
Extract: datashader-0.13.0-pyhd3eb1b0_1.conda
Extract: hvplot-0.7.3-pyhd3eb1b0_1.conda
Extract: anaconda-2022.05-py39_0.tar.bz2
Anaconda, Inc.
< Back
Next >
Cancel

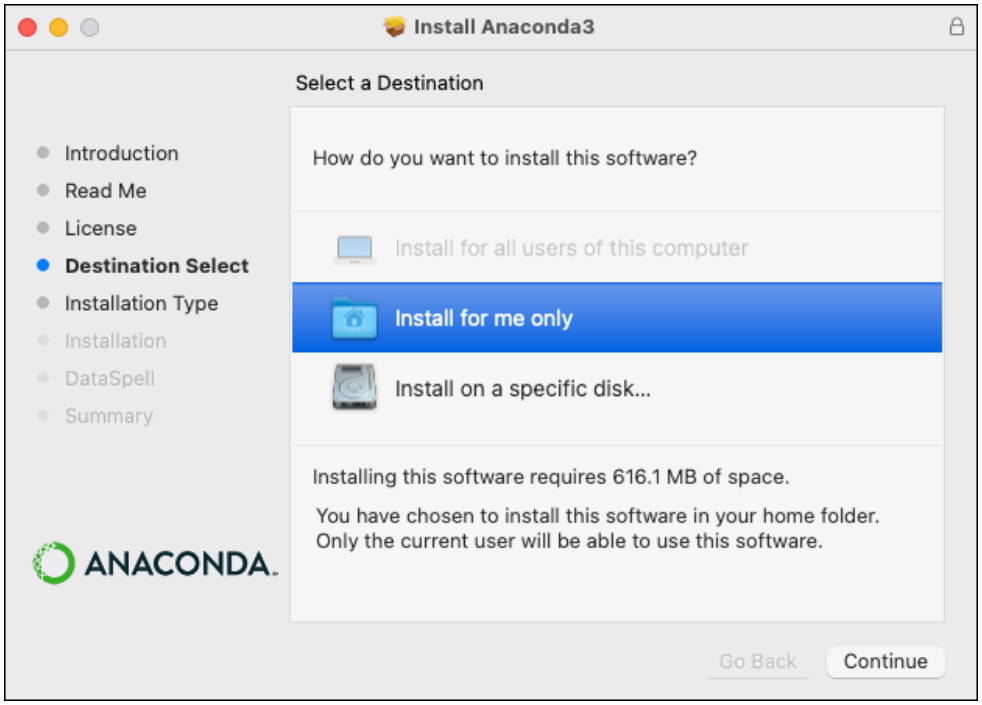

Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Installation Complete
Setup was completed successfully.
Completed
Processed C:\Users\DELL\anaconda3\Menu\console_shortcut.json successfully.
Processed C:\Users\DELL\anaconda3\Menu\notebook.json successfully.
Processed C:\Users\DELL\anaconda3\Menu\powershell_shortcut.json successfully.
Processed C:\Users\DELL\anaconda3\Menu\spyder_shortcut.json successfully.
Execute: "C:\Users\DELL\anaconda3\pythonw.exe" -E -s "C:\Users\DELL\anaconda3...
Running post install...
Execute: "C:\Users\DELL\anaconda3\pythonw.exe" -E -s "C:\Users\DELL\anaconda3...
Execute: "C:\Users\DELL\anaconda3\pythonw.exe" -E -s "C:\Users\DELL\anaconda3...
Created uninstaller: C:\Users\DELL\anaconda3\Uninstall-Anaconda3.exe
Completed
Anaconda, Inc.
< Back
Next >
Cancel


Anaconda3 2022.05 (64-bit) Setup
ANACONDA.
Anaconda3 2022.05 (64-bit)
Anaconda + JetBrains
Working with Python and Jupyter is a breeze in DataSpell. It is an IDE designed for exploratory data analysis and ML. Get better data insights with DataSpell.
DataSpell for Anaconda is available at:
https://www.anaconda.com/dataspell
ANACONDA.
DS
Anaconda, Inc.
< Back
Next >
Cancel



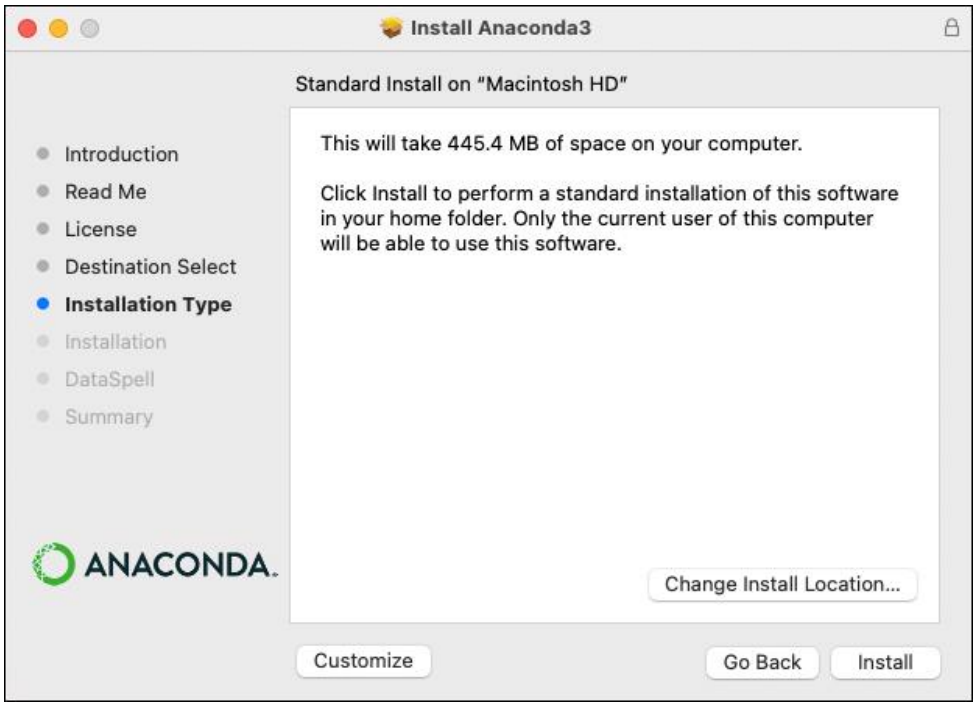
3 macOS Graphical Install:

1. Download the graphical macOS installer for your version of Python.
2. RECOMMENDED: Verify data integrity with SHA-256.
3. Double-click the downloaded file and click Continue to start the installation.
4. Answer the prompts on the Introduction, Read Me, and License screens.
5. You can click Change Install Location to install Anaconda Distribution for all users or on a specific disk. This is not recommended. Install for me only is the default (and recommended) selection. Click Continue to return to Installation Type.



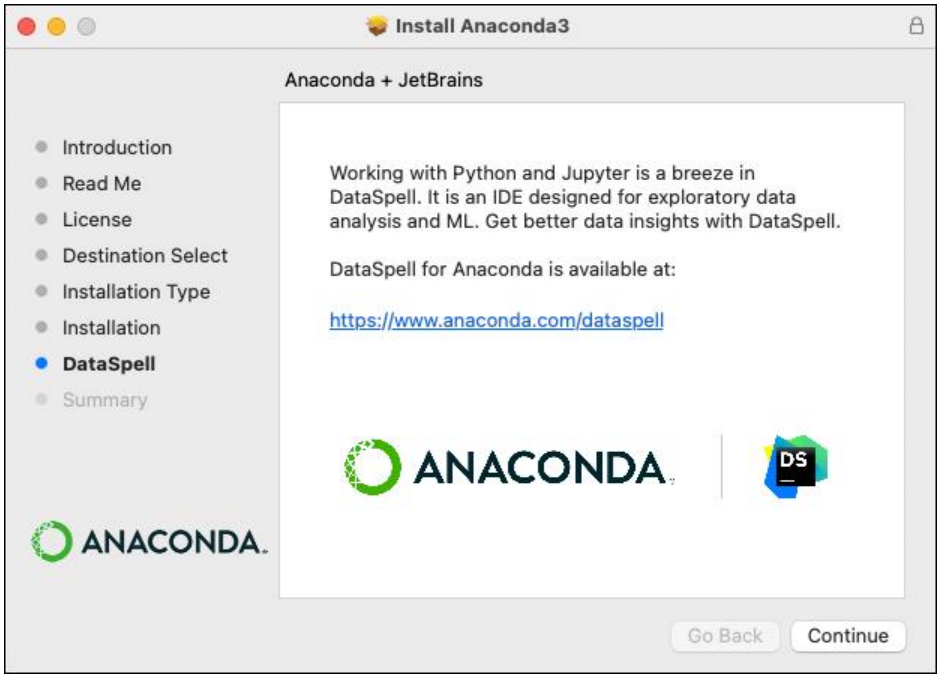
If you get the error message "You cannot install Anaconda in this location," reselect Install for me only.

6. Click Install to install Anaconda in your ~/opt directory (recommended).



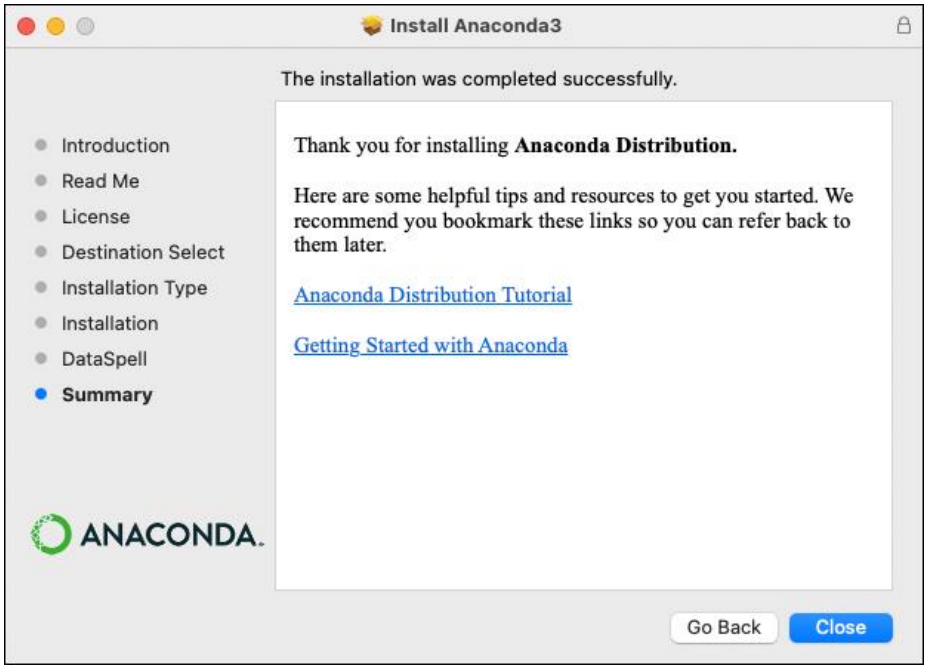
7. Once the install is complete, click Continue.

8. Optional: To install DataSpell for Anaconda, click https://www.anaconda.com/dataspell.



Or to continue without Dataspell, click Continue.

9. A successful installation displays the following screen:



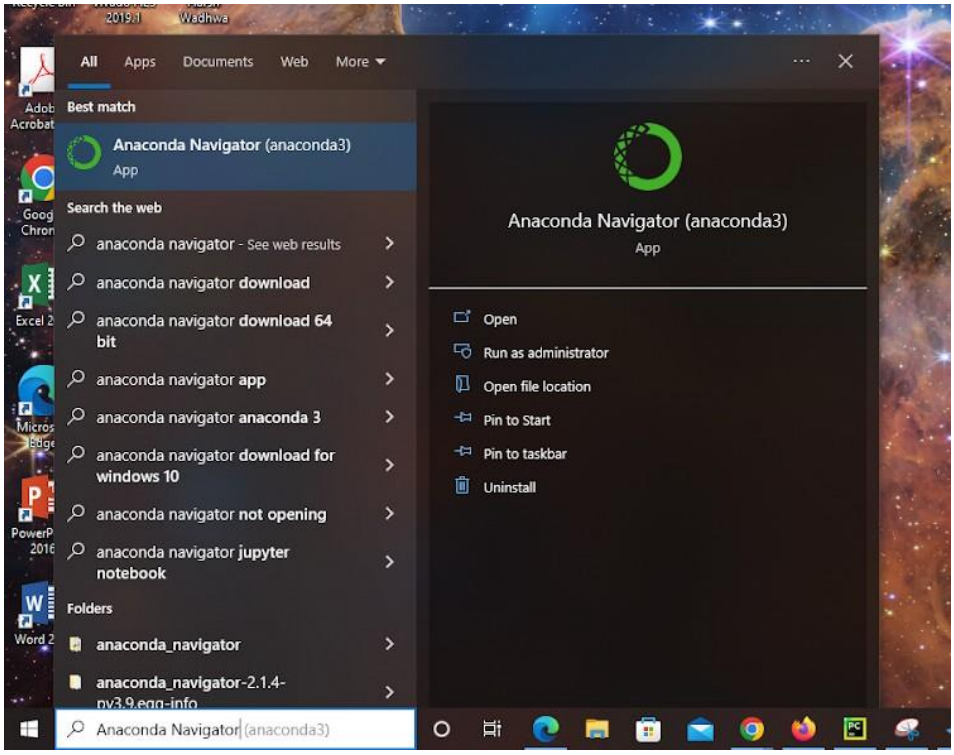
10. Verify your installation.

Creating the virtual environment “QCQML”

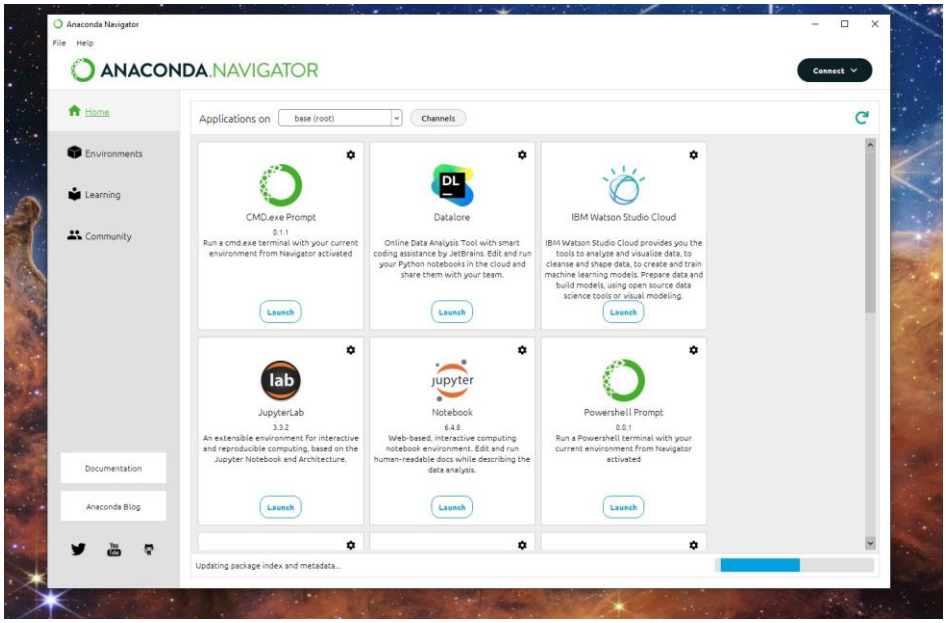
1. Start the Anaconda Navigator. In Windows you can find it in the Start Menu:

To try Navigator, after installing Anaconda, on other operating systems, click the Navigator icon on your operating system’s program menu, or in Anaconda prompt (or terminal on Linux or macOS), run the command

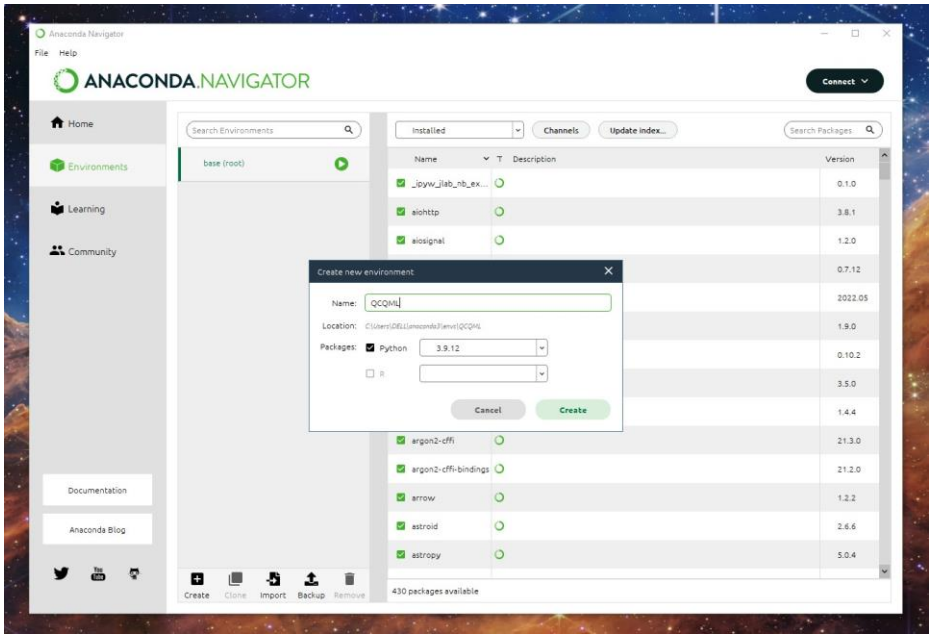
```
(environment_name) ...      $ anaconda-navigator
```



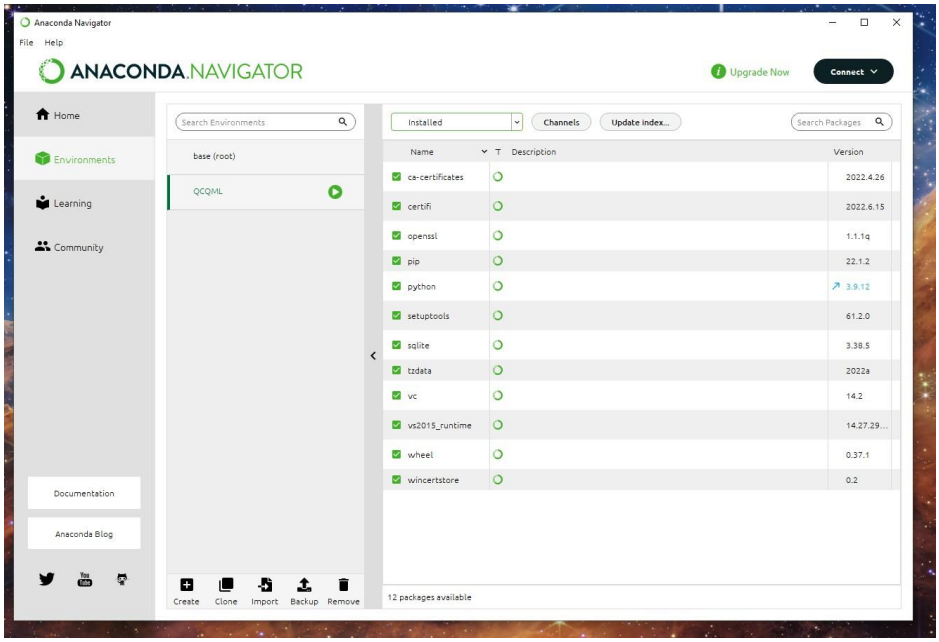
2. Once started, Anaconda Navigator will show its Home tab.



3. Select the Environments tab on the left. And click on the Create button on the bottom. Fill in "QCQML" in the environment name and click Create.

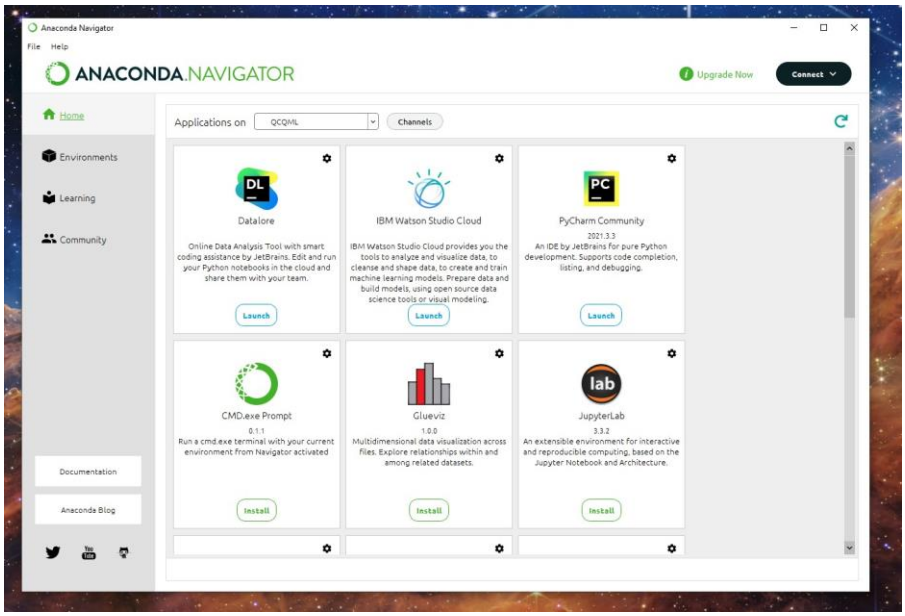


It'll take a while until the creation is completed. Once ready, you'll see a new environment "QCQML" in the middle pane of Anaconda Navigator:



Installing Jupyter Notebook in the “QCQML” environment

1. Go back to the Home tab. Make sure that “QCQML” is selected in the drop-down menu where it says “Applications on.” If everything looks OK, click the Install button on the Jupyter Notebook card.

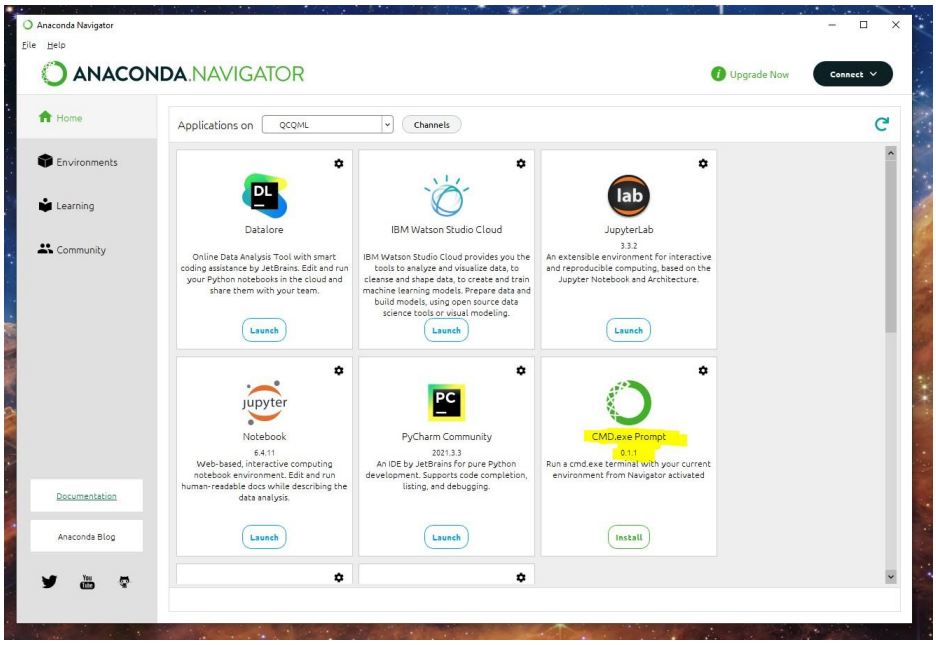


It will take a while to complete the installation.

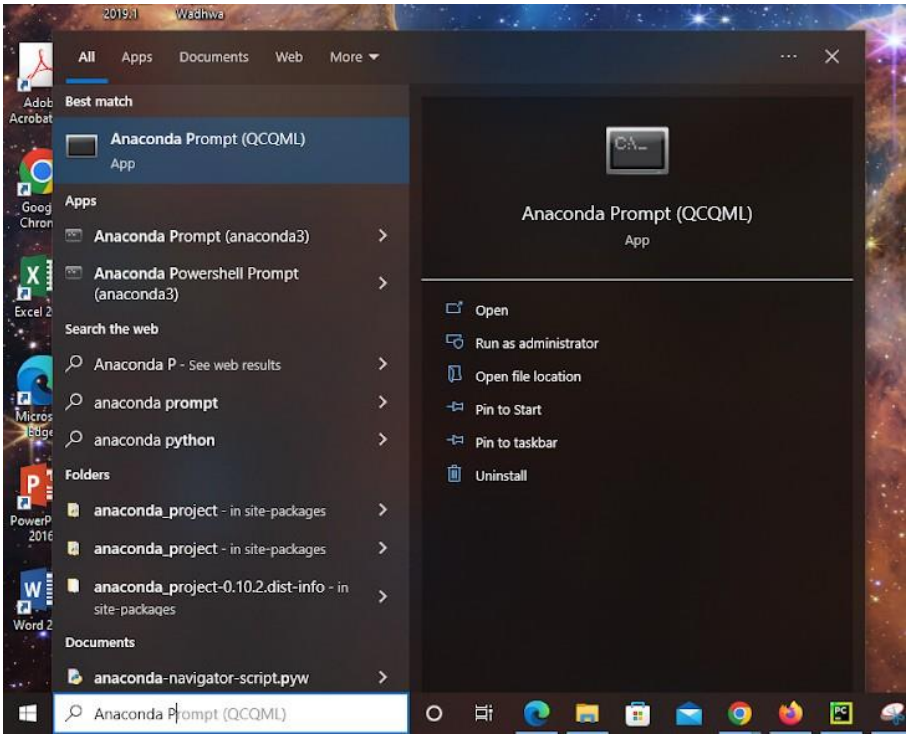
Enabling the Anaconda Terminal in the "QCQML" environment (optional)

You should be able to install all dependencies and use the materials from Jupyter Notebook alone, however, you might find it useful to have the means to manage the "QCQML" virtual environment from the command line.

1. Select the Home tab, and make sure that the "QCQML" environment is selected in the drop-down menu on the top. Find the card "CMD.exe Prompt" and click "Install:"



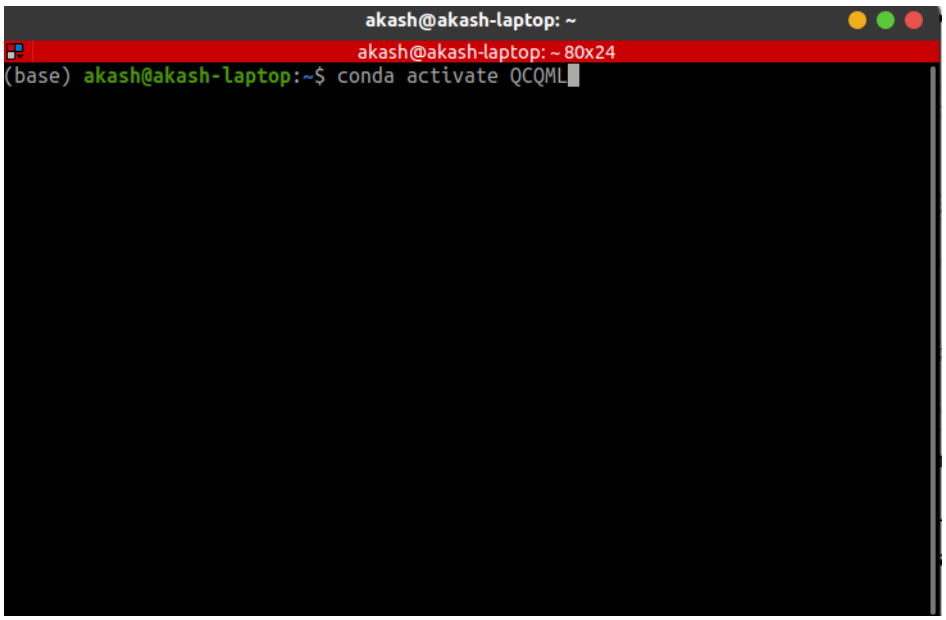
After this, a new menu item, Anaconda Prompt (QCQML) will be available in the Anaconda3 folder in the Start Menu:

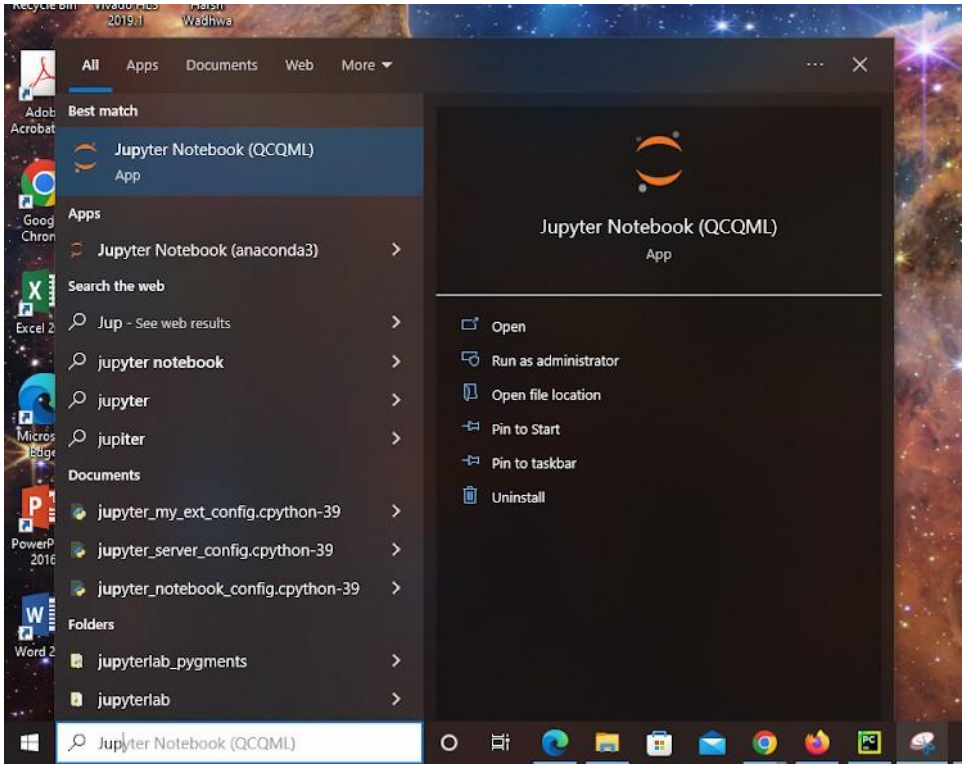


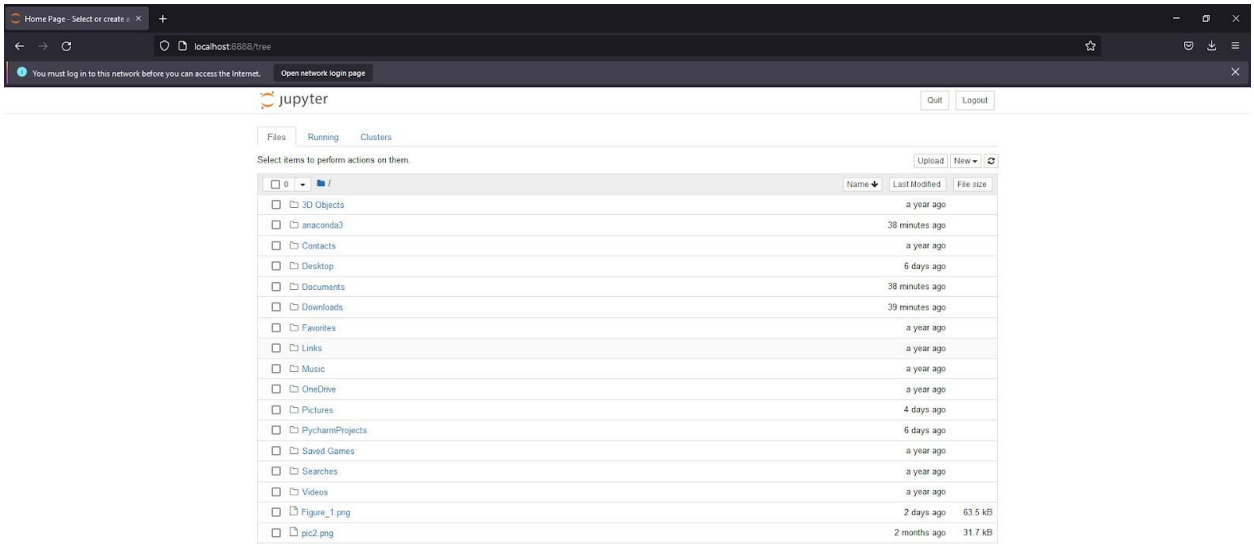
For Mac and Linux: Open the terminal and enter

```
(environment_name) ... $ conda activate QCQML
```

This will change your environment from base to QCQML







Starting Jupyter Notebook

We recommend starting Jupyter Notebook in a separate virtual environment named "QCQML"

1. If you followed the instructions above, a new item "Jupyter Notebook (QCQML)" will be available in Start Menu under the Anaconda3 folder: You can also launch the "Jupyter Notebook" from Anaconda Navigator.

Qiskit Installation

The official installation guide is https://docs.quantum.ibm.com/start/install

/ For Windows: Open Anaconda Prompt (QCQML) (Windows users need not use "$")

```
(QCQML) … $ pip install qiskit

(QCQML) … $ pip install qiskit-terra[visualization]

(QCQML) … $ pip install qiskit-ibm-runtime

(QCQML) … $ pip install qiskit-aer
```

For Mac and Linux: Open the terminal and enter

```
(environment_name) ...  $ conda activate QCQML
```

And follow the same steps as above.

Keeping your Qiskit environment up to date

Qiskit is an open-source programming environment that is in continuous flux.

It is generally a good idea to stay updated with the latest version, but with some updates, components of the code might change behavior. It is always a good idea to have a good look at the release notes for each new version. Sometimes changes are introduced that will change the way your code behaves. In those cases, you might want to hold off on upgrading until you have verified that your code still works as expected.

If you are using Anaconda environments, then you can maintain more than one environment at different Qiskit levels, to have a fallback environment in case an upgraded Qiskit version breaks your code.

Getting ready

Before you begin, verify which version of Qiskit you are running for each of your environments (if you have more than one).
For each environment, launch Python, either from the command line, from an IDE such as Spyder, or as a Jupyter notebook, then execute the following code:

```
        $ python3
>>> import qiskit
>>> qiskit.__qiskit_version__
```

If you have an old version of Qiskit installed, the preceding code might result in the following output:

```
        {'qiskit-terra': '0.20.2', 'qiskit-aer': '0.10.4', 'qiskit-ibmq-
        provider': '0.19.1', 'qiskit-aqua': '0.9.5', 'qiskit': '0.36.2'}
```

You can then go to the Qiskit release notes to find out if there is a more up-to-date version available: https://qiskit.org/documentation/release_notes.html

How to do it

1. Activate your virtual environment:

```
 (environment_name) …  $ conda activate QCQML
```

2. Run the following command to check for outdated pip packages for your virtual Environment:

```
(QCQML) ... $ pip list --outdated
```

3. This will return a list of all your pip packages that are currently outdated and list the available versions

4. It is then a breeze to update Qiskit using pip:

```
(QCQML) ... $ pip install qiskit --upgrade
```

5. Verify that Qiskit is integrated with Python in your isolated environment.
Open Python:

```
(QCQML)...      $      python3
```

Import Qiskit®:

```
>>>   import   qiskit
```

List the version details:

```
>>> qiskit.__qiskit_version__
```

This should display the versions of the installed Qiskit components.
Congratulations, your Qiskit upgrade worked; you are now running the latest code!

Creating your IBM Quantum Experience account

Your key to exploring quantum computer programming with IBM is your IBM Quantum Experience account. This free account gives you access to the online IBM Quantum Experience interface, and the programming tools that are available there. An IBM Quantum Experience account is not technically required to test out IBM Quantum Experience or to install Qiskit but is required to run your programs on the freely available IBM quantum computers.

Getting ready

To set up your IBM Quantum Experience account, you can log in with an IBMid, or with one of the following:

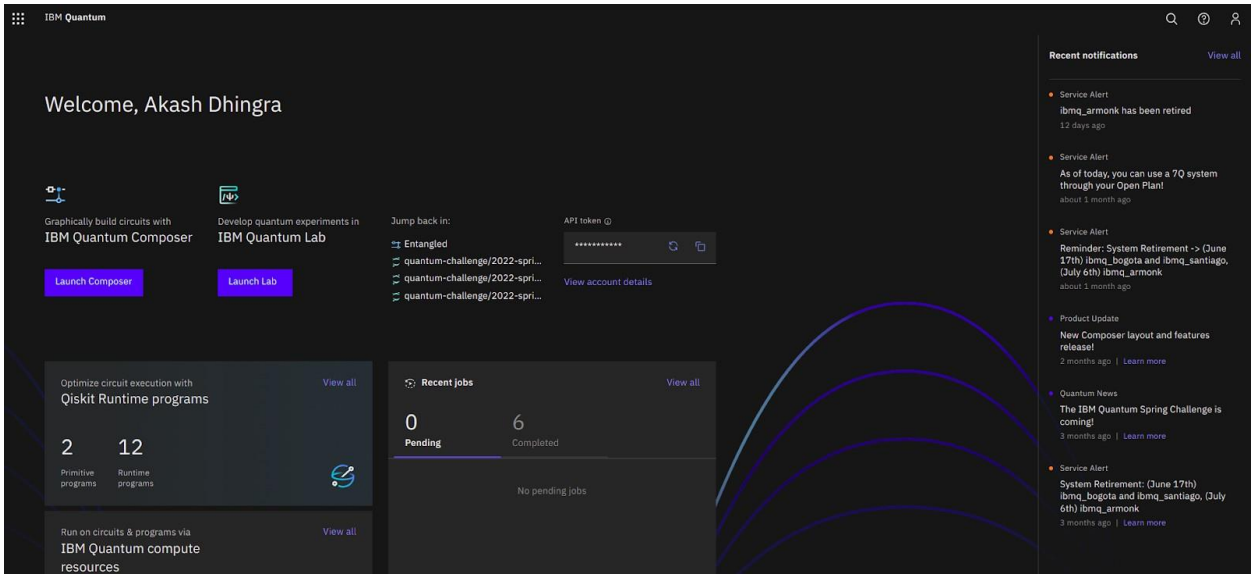
- A Google account
- A GitHub account
- A LinkedIn account
- A Twitter account
- An email address

How to do it...

1. In your browser (Google Chrome seems to work best), go to this link: https://quantum-computing.ibm.com/login.

2. Enter your IBMid credentials or select another login method.
You can also skip the sign-in, which will give you access to IBM Quantum Experience but with a limit of 3 qubits for your quantum circuits, and with simulator backends only.

3. Once you have logged in, you now have an activated IBM Quantum Experience account, and will find yourself at the main dashboard:



Cirq Installation

The official installation guide is https://quantumai.google/cirq/start/install

Installing on Linux

1. Make sure you have python 3.7.0 or greater.
   See Installing Python 3 on Linux @ the hitchhiker's guide to python.
2. Consider using a virtual environment.

   Open terminal:

   ```
   (environment_name) ...      $ conda activate QCQML
   ```

3. Use pip to install cirq:

   ```
   (QCQML) ... $ python -m pip install --upgrade pip
   ```

   ```
   (QCQML) ... $ python -m pip install cirq
   ```

4. (Optional) install other dependencies.
   Install dependencies of features in cirq.contrib.

```
(QCQML) … $ python -m pip install cirq-core[contrib]
```

5. Install system dependencies that pip can't handle.

```
(QCQML) … $ sudo apt-get install texlive-latex-base latexmk
```

   - Without texlive-latex-base and latexmk, pdf writing functionality will not work.

6. Check that it works!

```
(QCQML) … $ python -c 'import cirq_google; print(cirq_google.Sycamore)'
```

```
# should print:
#                          (0, 5)———(0, 6)
#                            |       |
#                            |       |
#                    (1, 4)———(1, 5)———(1, 6)———(1, 7)
#                      |       |       |       |
#                      |       |       |       |
#                (2, 3)———(2, 4)———(2, 5)———(2, 6)———(2, 7)———(2, 8)
#                  |       |       |       |       |       |
#                  |       |       |       |       |       |
#          (3, 2)———(3, 3)———(3, 4)———(3, 5)———(3, 6)———(3, 7)———(3, 8)———(3, 9)
#            |       |       |       |       |       |       |   |
#            |       |       |       |       |       |       |   |
#      (4, 1)———(4, 2)———(4, 3)———(4, 4)———(4, 5)———(4, 6)———(4, 7)———(4, 8)———(4, 9)
#        |       |       |       |       |       |       |   |
#        |       |       |       |       |       |       |   |
# (5, 0)———(5, 1)———(5, 2)———(5, 3)———(5, 4)———(5, 5)———(5, 6)———(5, 7)———(5, 8)
#        |       |       |       |       |       |   |
#        |       |       |       |       |       |   |
#      (6, 1)———(6, 2)———(6, 3)———(6, 4)———(6, 5)———(6, 6)———(6, 7)
#        |       |       |       |   |
#        |       |       |       |   |
#          (7, 2)———(7, 3)———(7, 4)———(7, 5)———(7, 6)
#            |       |       |
#            |       |       |
#                (8, 3)———(8, 4)———(8, 5)
#                  |
#                  |
#                (9, 4)
```

Installing on Mac OS X

1. Make sure you have python 3.7.0 or greater.
   See Installing Python 3 on Mac OS X @ the hitchhiker's guide to python.
2. Consider using a virtual environment.

   Open Teminal:

   ```
   (environment_name) … $ conda activate QCQML
   ```

3. Use pip to install cirq:

   ```
   (QCQML) … $ python -m pip install --upgrade pip
   ```

   ```
   (QCQML) …  $ python -m pip install cirq
   ```

4. (Optional) install dependencies of features in cirq.contrib.

   ```
   (QCQML) … $ python -m pip install cirq-core[contrib]
   ```

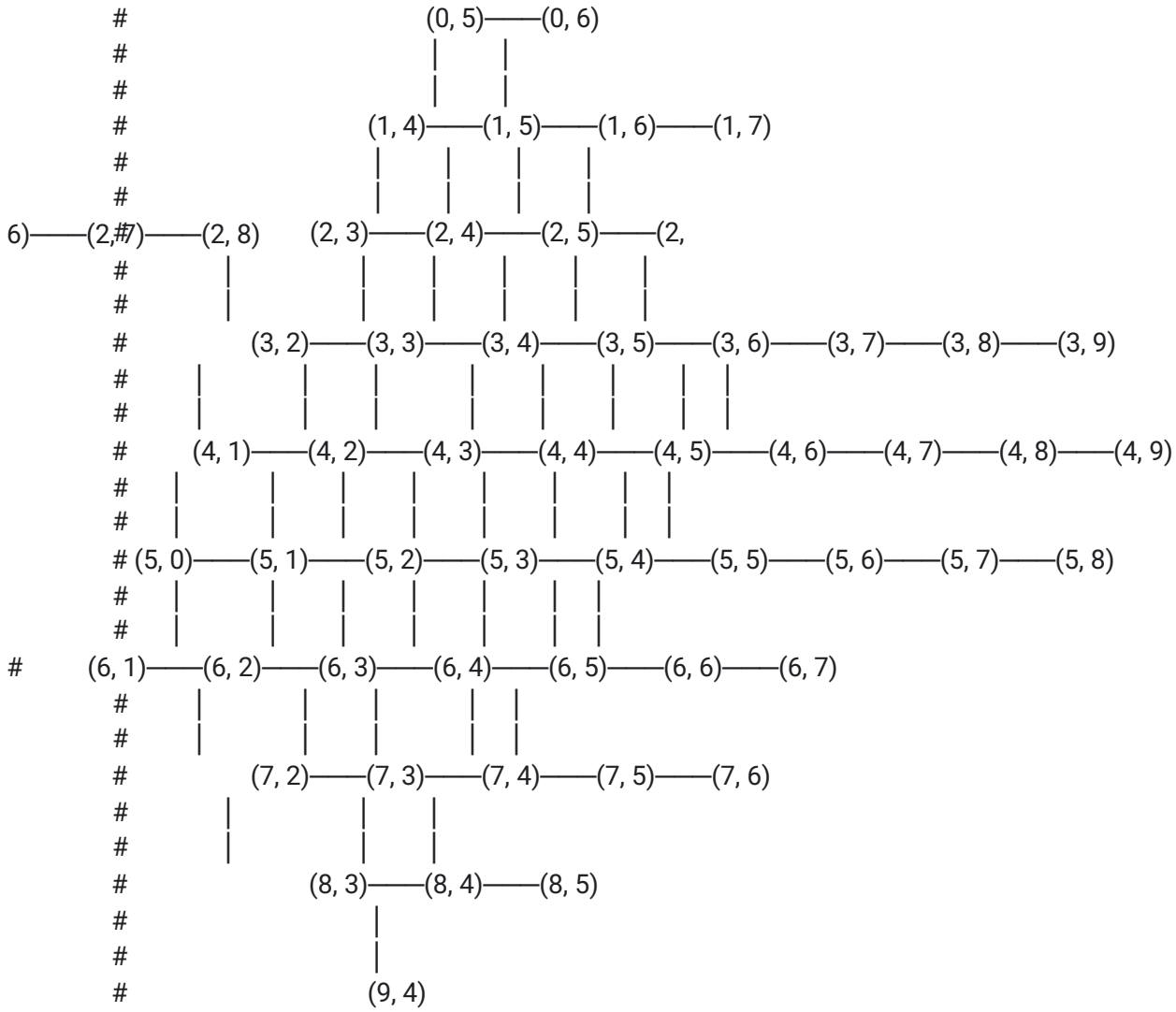
5. Install system dependencies that pip can't handle.

   ```
   (QCQML) … $ brew cask install mactex
   ```

   - Without mactex, pdf writing functionality will not work.
6. Check that it works!

   ```
   (QCQML) … $ python -c 'import cirq_google; print(cirq_google.Sycamore)' #
   ```

   should print:



Installing on Windows

1. If you are using the Windows Subsystem for Linux, use the Linux install instructions instead of these instructions.
2. Make sure you have python 3.7.0 or greater.
   See Installing Python 3 on Windows @ the hitchhiker's guide to python.
3. Open Anaconda Prompt (QCQML) from Start Menu
4. Use pip to install cirq:

   ```
   (QCQML) ... > python -m pip install --upgrade pip
   ```

   ```
   (QCQML) ... > python -m pip install cirq
   ```

5. (Optional) install dependencies of features in cirq.contrib.

```
(QCQML) ... > python -m pip install cirq-core[contrib]
```

6. Check that it works!

```
(QCQML) ... > python -c "import cirq_google; print(cirq_google.Sycamore)"
```

```
# should print:

#                          (0, 5)———(0, 6)
#                            |       |
#                            |       |
#                        (1, 4)———(1, 5)———(1, 6)———(1, 7)
#                          |       |       |       |
#                          |       |       |       |
#                    (2, 3)———(2, 4)———(2, 5)———(2, 6)———(2, 7)———(2, 8)
#                      |       |       |       |       |       |
#                      |       |       |       |       |       |
#              (3, 2)———(3, 3)———(3, 4)———(3, 5)———(3, 6)———(3, 7)———(3, 8)———(3, 9)
#        |           |       |       |       |       |       |   |
#        |           |       |       |       |       |       |   |
#       (4, 1)———(4, 2)———(4, 3)———(4, 4)———(4, 5)———(4, 6)———(4, 7)———(4, 8)———(4, 9)
#     |           |       |       |       |       |       |   |
#     |           |       |       |       |       |       |   |
# (5, 0)———(5, 1)———(5, 2)———(5, 3)———(5, 4)———(5, 5)———(5, 6)———(5, 7)———(5, 8)
#     |           |       |       |       |       |   |
#     |           |       |       |       |       |   |
#       (6, 1)———(6, 2)———(6, 3)———(6, 4)———(6, 5)———(6, 6)———(6, 7)
#        |           |       |         |   |
#        |           |       |         |   |
#              (7, 2)———(7, 3)———(7, 4)———(7, 5)———(7, 6)
#          |             |       |
#          |             |       |
#                    (8, 3)———(8, 4)———(8, 5)
#                      |
#                      |
#                    (9, 4)
```