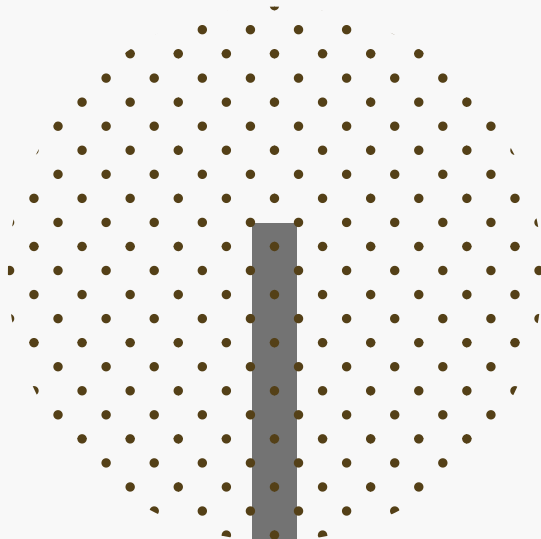


INSTALLATION

QSILVER



Welcome to QSilver!

This workshop consists of a collection of Jupyter notebooks

We use Python 3 (3.6+) as our programming language and we will be working on libraries Qiskit and Cirq!

This is a beginners guide to install Qiskit and Cirq!

For a video walkthrough of the installation please check this [video](#)

Note: If you have already installed Qiskit or Cirq on your system, you may skip this guide and move on to the Start notebook

If you have Jupyter notebook/lab already installed, you may skip to the installation notebooks section for the installation part

Installing Anaconda®



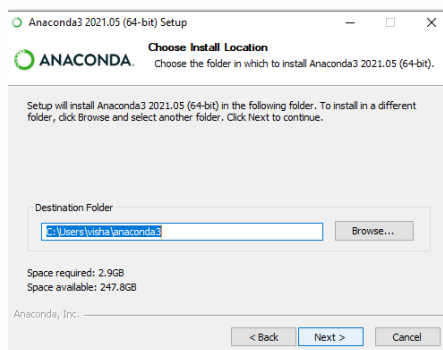
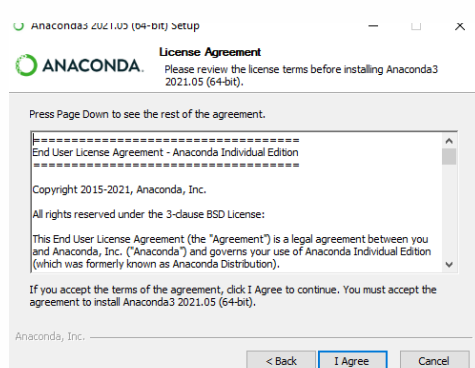
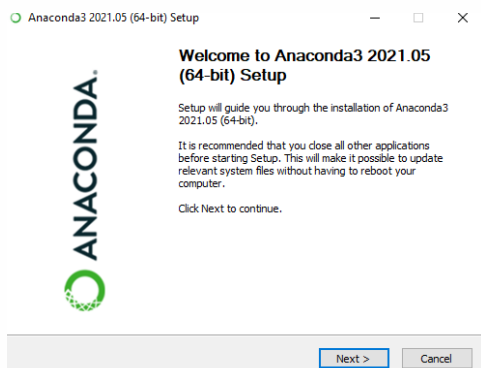
Installing Anaconda® will install the required Python compiler and libraries you need with one click! It will also make it easy to install Jupyter notebooks

[Click here to Download Anaconda](#)

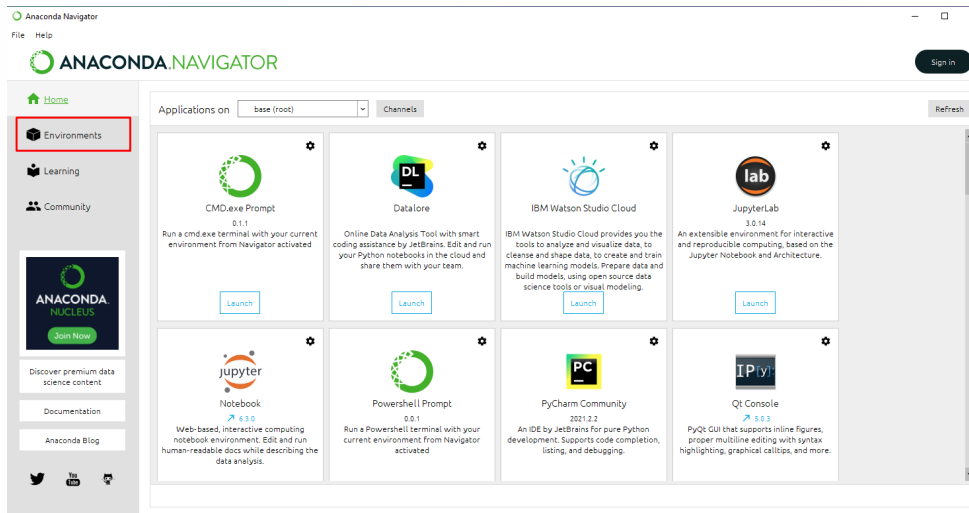
Setting up Anaconda®



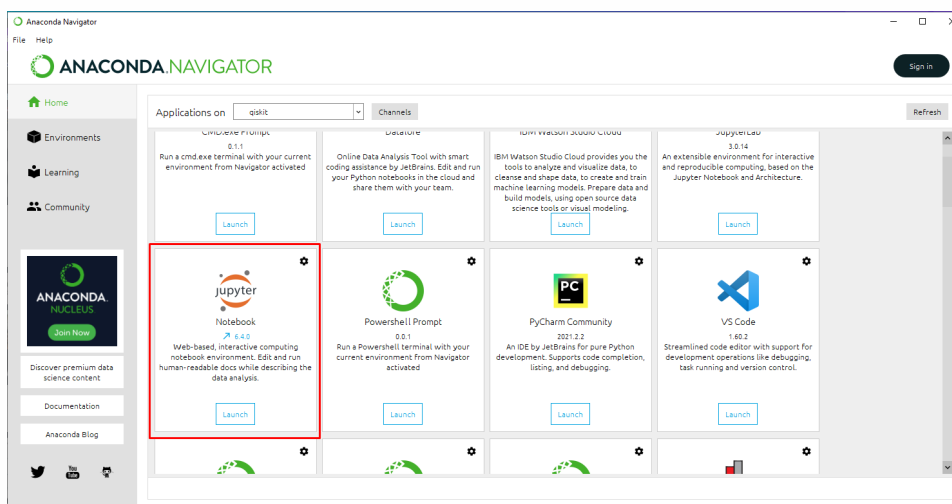
Select your preferred operating system and download the setup file. Run the setup file after downloading



Setting up Anaconda®



In the Annaconda navigator you can either directly install jupyter as shown below or you can create a separate new environment for this task

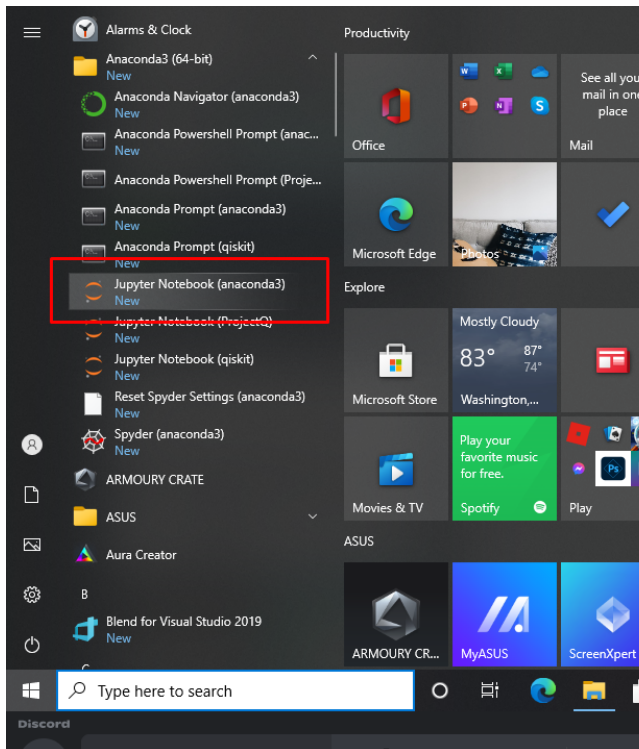


Proceed to install jupyter from the navigator dashboard



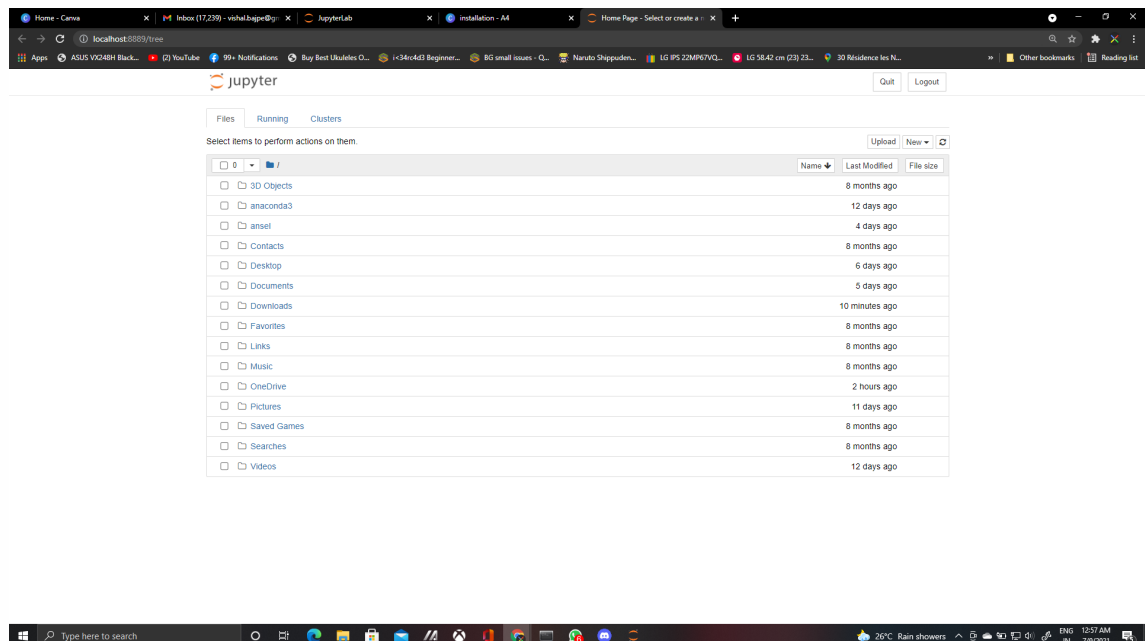
We are working to build an open quantum ecosystem
More information: qworld.net

Setting up Jupyter notebooks



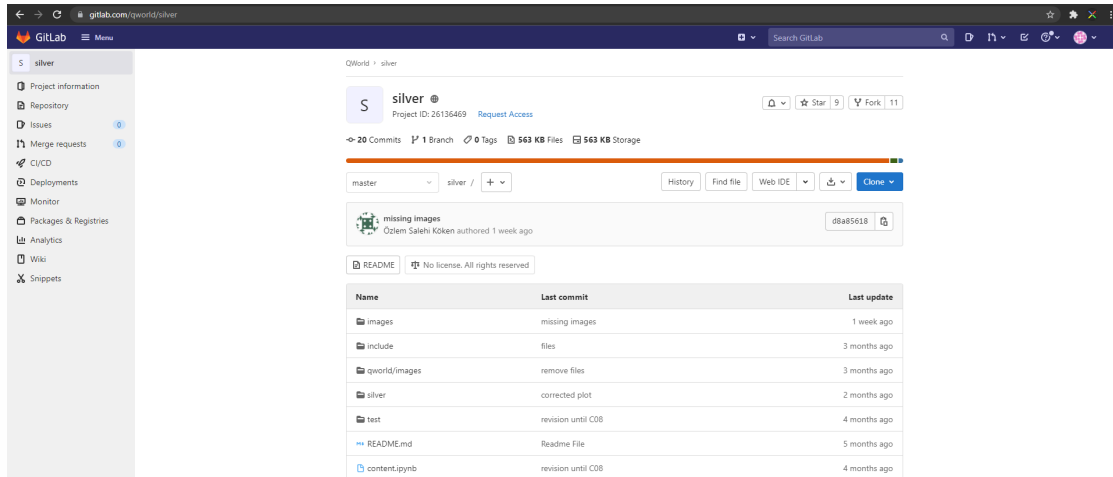
Run jupyter notebook from the start menu. A command prompt or powershell window should appear following an opening of a web browser page that looks like one below.

Note: **Keep the command prompt window open while working.**



Setting up QSilver files

Download the QSilver files from this [link](#)



Extract the files to one of the accessible directories e.g. Desktop, Documents, Downloads etc.

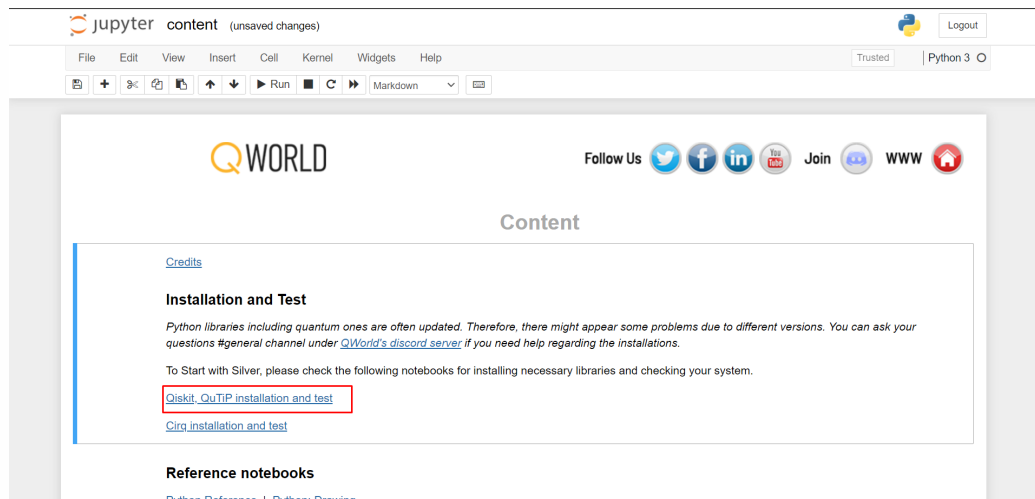
We have extracted our files in the Desktop here and we can access it from the dashboard as shown



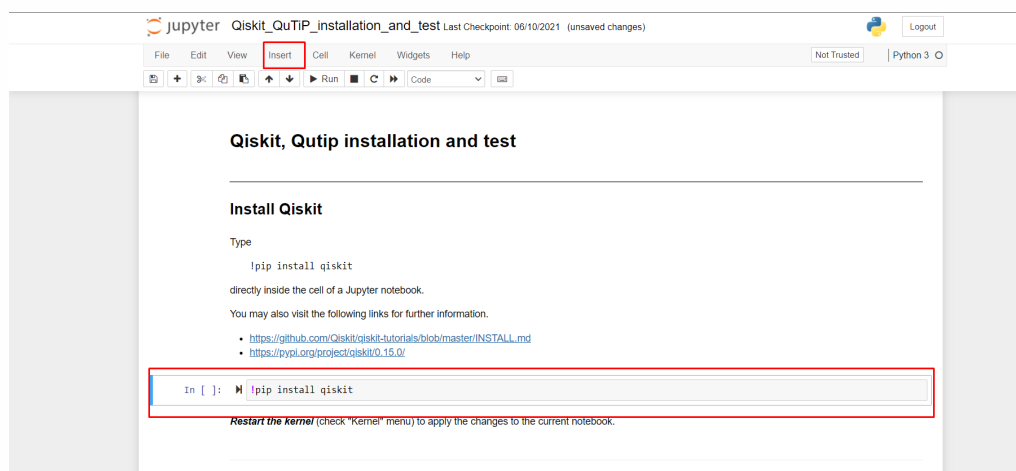
We are working to build an open quantum ecosystem
More information: qworld.net

Installing Qiskit and Qutip

1) Open the notebook "content.ipynb" from the dashboard and open "Qiskit, QuTiP installation and test"

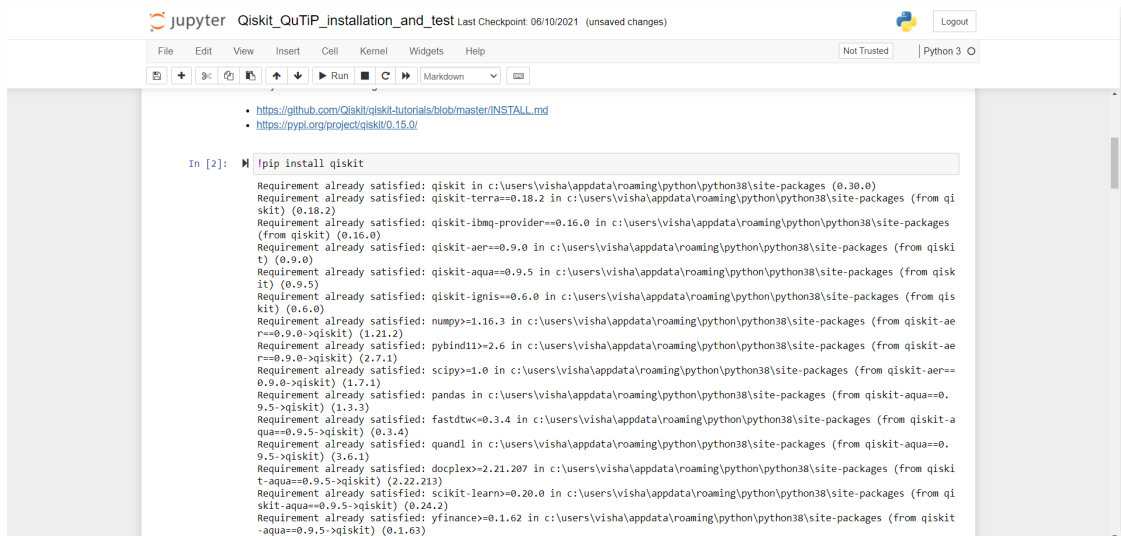


2) Create a new cell and type in "!pip install qiskit" to initiate the installation. *To create a new cell, click on the Insert tab on top. To run the cell, click on the Run button on top left*



Installing Qiskit and Qutip

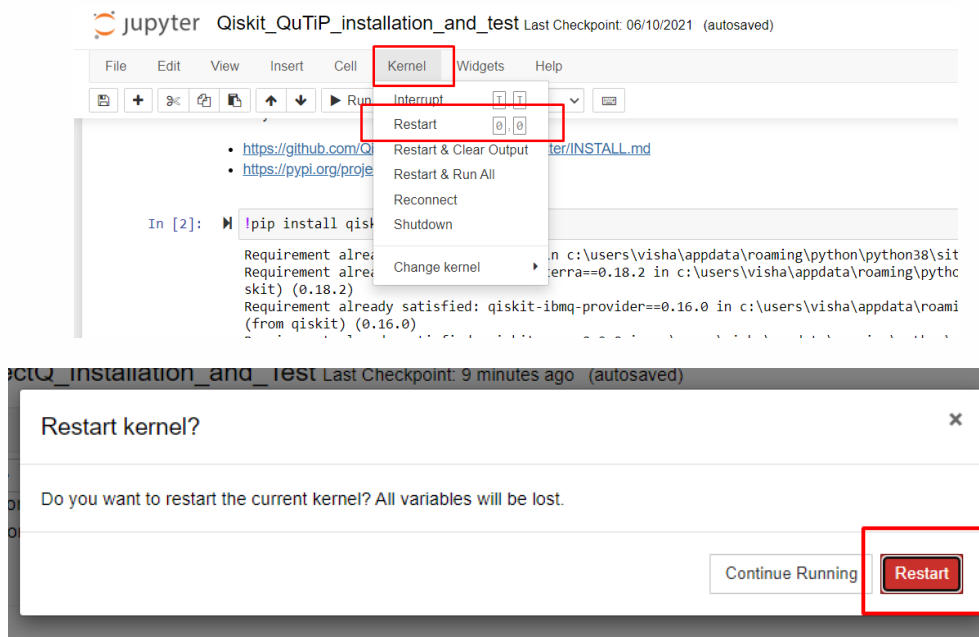
3) Run this cell to initiate installation of Qiskit



```
In [2]: !pip install qiskit

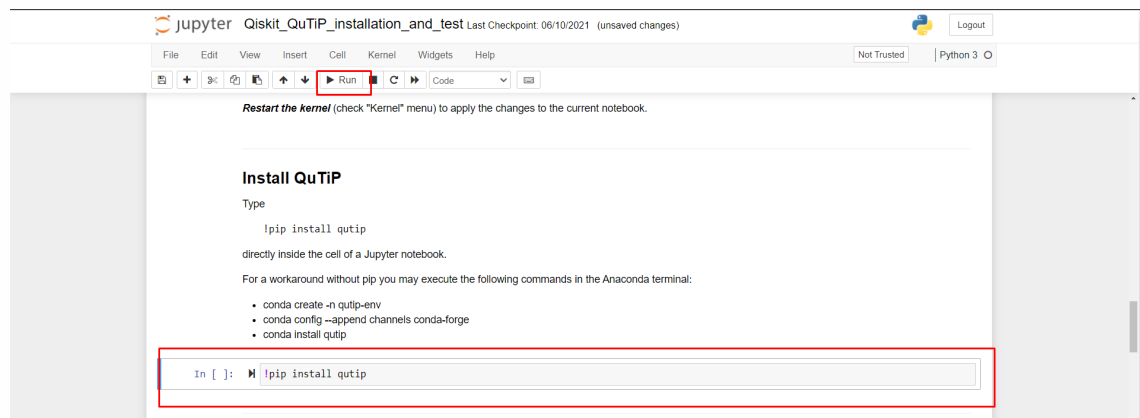
Requirement already satisfied: qiskit in c:\users\visha\appdata\roaming\python\python38\site-packages (0.30.0)
Requirement already satisfied: qiskit-terra==0.18.2 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit) (0.18.2)
Requirement already satisfied: qiskit-ibmq-provider==0.16.0 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit) (0.16.0)
Requirement already satisfied: qiskit-aer==0.9.0 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit) (0.9.0)
Requirement already satisfied: qiskit-aqua==0.9.5 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit) (0.9.5)
Requirement already satisfied: qiskit-ignis==0.6.0 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit) (0.6.0)
Requirement already satisfied: numpy>=1.16.3 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aer==0.9.0->qiskit) (1.21.2)
Requirement already satisfied: pybind11>=2.6 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aer==0.9.0->qiskit) (2.7.1)
Requirement already satisfied: scipy>=1.0 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aer==0.9.0->qiskit) (1.7.1)
Requirement already satisfied: pandas in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aqua==0.9.5->qiskit) (1.3.3)
Requirement already satisfied: fastdtw<0.3.4 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aqua==0.9.5->qiskit) (0.3.4)
Requirement already satisfied: quandl in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aqua==0.9.5->qiskit) (3.6.1)
Requirement already satisfied: docplex>=2.21.207 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aqua==0.9.5->qiskit) (2.22.213)
Requirement already satisfied: scikit-learn>=0.20.0 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aqua==0.9.5->qiskit) (0.24.2)
Requirement already satisfied: yfinance>=0.1.62 in c:\users\visha\appdata\roaming\python\python38\site-packages (from qiskit-aqua==0.9.5->qiskit) (0.1.63)
```

4) If there are no errors, Restart your kernel to finalize installation



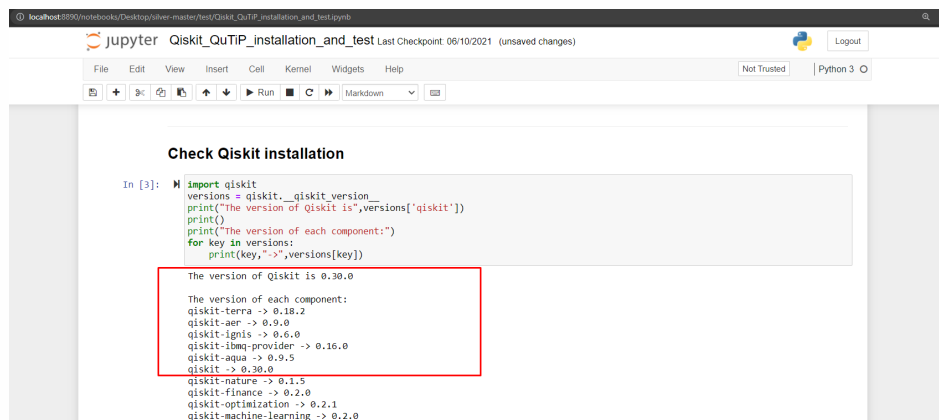
Installing Qiskit and Qutip

5) Let us now install Qutip. Proceed by creating a new cell and typing in "!pip install qutip" and run the cell. Restart the kernel as shown in Step 4 to finalize the installation



Testing Qiskit and Qutip installation

1) Run the cell below to check the Qiskit installation. You should get an output showing the components installed and their versions.



Testing Qiskit and Qutip installation

2) Run the cells preceding that. You should notice circuit outputs and the Bloch sphere diagram to indicate successful installation of Qiskit and Qutip!

```
3) Execute the circuit 1024 times in the local simulator and print the observed outcomes

In [3]: M ## execute the circuit 1024 times
job = execute(circuit, Aer.get_backend('qasm_simulator'), shots=1024)
# get the result
counts = job.result().get_counts(circuit)
print(counts)

{'01': 583, '10': 521}

4) Check Qutip - draw a Bloch sphere

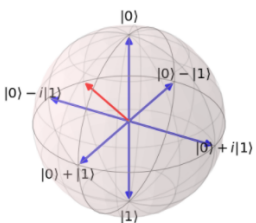
In [5]: M from qiskit.visualization import plot_bloch_vector, bloch
from matplotlib.pyplot import text
from math import pi, cos, sin
from qutip import *

theta = pi/2
phi = 4*pi/3

x = sin(theta)*cos(phi)
y = sin(theta)*sin(phi)
z = cos(theta)

sphere = bloch
sphere.Bloch
b = Bloch()
b.ylpos = [1.1, -1.2]
b.xlabel = ['$\\left|0\\right\\rangle+\\left|1\\right\\rangle$', '$\\left|0\\right\\rangle-\\left|1\\right\\rangle$']
b.ylabel = ['$\\left|0\\right\\rangle+i\\left|1\\right\\rangle$', '$\\left|0\\right\\rangle-i\\left|1\\right\\rangle$']
b.vector_color = ['b','b','b','b','b','r']
b.add_vectors([[0,0,1],[0,0,-1],[0,1,0],[0,-1,0],[1,0,0],[-1,0,0]])
b.add_vectors([x,y,z])
b.show()

# re-execute this cell if you DO NOT see the Bloch sphere
```



3) Let us now move on to Cirq installation! Exit this notebook and head back to content.py. Click on the "Cirq installation and test" to proceed

Installation and Test

Python libraries including quantum ones are often updated. Therefore, there might appear some problems due to different versions. You can ask your questions #general channel under [QWorld's discord server](#) if you need help regarding the installations.

To Start with Silver, please check the following notebooks for installing necessary libraries and checking your system.

[Qiskit, QuTIP installation and test](#)

[Cirq installation and test](#)



We are working to build an open quantum ecosystem
More information: qworld.net

Installing Cirq

1) Let us setup and install cirq now. Proceed by creating a new cell and typing in "!pip install cirq"

The screenshot shows a Jupyter notebook titled "Cirq installation and test" within the QWorld environment. The notebook contains a code cell with the command `!pip install cirq`. A red box highlights this code cell. Below the code cell, a message reads: "Restart the kernel (check 'Kernel' menu) to apply the changes to the current notebook." The notebook also includes a header with the QWorld logo and a note about macros.

2) Restart the kernel to finalize the installation

This block contains two screenshots. The top screenshot shows the Jupyter notebook's "Kernel" menu with the "Restart the kernel" option highlighted. The bottom screenshot shows a dialog box titled "Restart kernel?" with the text "Do you want to restart the current kernel? All variables will be lost." The "Restart" button is highlighted with a red box.

Testing Cirq

1) Run this cell to test out the cirq installation and print the map layout of Google's Foxtail chip

Check Cirq installation

```
In [3]: import cirq

print(cirq.google.Foxtail)
# should print:
# (0, 0)---(0, 1)---(0, 2)---(0, 3)---(0, 4)---(0, 5)---(0, 6)---(0, 7)---(0, 8)---(0, 9)---(0, 10)
#
# (1, 0)---(1, 1)---(1, 2)---(1, 3)---(1, 4)---(1, 5)---(1, 6)---(1, 7)---(1, 8)---(1, 9)---(1, 10)
#
# (0, 0)---(0, 1)---(0, 2)---(0, 3)---(0, 4)---(0, 5)---(0, 6)---(0, 7)---(0, 8)---(0, 9)---(0, 10)
#
# (1, 0)---(1, 1)---(1, 2)---(1, 3)---(1, 4)---(1, 5)---(1, 6)---(1, 7)---(1, 8)---(1, 9)---(1, 10)
```

2) Run the preceding cells to execute a sample circuit and test out cirq

Execute An Example Program

1) Create a quantum circuit

```
In [2]: import cirq

# Pick a qubit.
qubit = cirq.GridQubit(0, 0)

# Create a circuit
circuit = cirq.Circuit(
    cirq.X(qubit)**0.5, # Square root of NOT.
    cirq.measure(qubit, key='m') # Measurement.
)
```

2) Draw the circuit

Run the cell once more if the figure is not shown

```
In [3]: print("Circuit:")
print(circuit)

Circuit:
(0, 0): —X^0.5—M('m')—
```

3) Execute the circuit 20 times in the local simulator and print the observed outcomes

```
In [4]: # Simulate the circuit several times.
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=20)
print("Results:")
print(result)

Results:
m=01010000100001100010
```

4) Print a histogram of results

```
In [5]: # Print a histogram of results
results = result.histogram(key='m')
```

If there are no errors, your installation is complete and successful! You can continue onto the further notebooks in the "content" section and start your Silver journey! :D

Prepared by - Vishal Sharathchandra Bajpe



We are working to build an open quantum ecosystem
More information: qworld.net