We are going to import everything from qiskit

In [15]:
```
from qiskit import *
from qiskit import IBMQ
```

In [16]:
```
IBMQ.save_account('64fab657c85c2cc10c6040d7f0b08ba1224cd937bd2cc461cc279f3c4d6e81864e19cbd9de0ebafd2a825a4ab2970312e94cd46e21389cb0ab642f67f388ed01')
```

```
configrc.store_credentials:WARNING:2022-01-04 21:59:51,104: Credentials already present. Set overwrite=True to overwrite.
```

In [17]:
```
IBMQ.load_account()
```

```
ibmqfactory.load_account:WARNING:2022-01-04 21:59:51,971: Credentials are already in use. The existing account in the session will be replaced.
```

Out[17]:
```
<AccountProvider for IBMQ(hub='ibm-q', group='open', project='main')>
```

In [18]:
```
import numpy as np
```

We are going to register quantum bits and classical bits

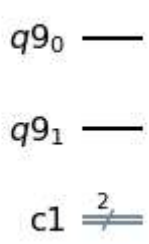
In [19]:
```
qr = QuantumRegister(2)
cr = ClassicalRegister(2)
```

Let's define our quantum circuit

In [20]:
```
circuit = QuantumCircuit(qr,cr)
```

In [21]:
```
circuit.draw(output='mpl')
```

Out[21]:

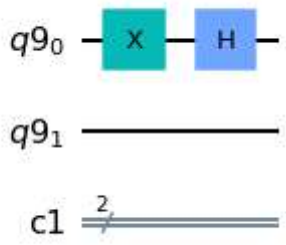


In [22]:
```
circuit.x(0)
circuit.h(0)
```

Out[22]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x1f42c030778>
```

In [23]:
```
circuit.draw(output='mpl')
```

Out[23]:



Let us see the statevector representation

In [24]:
```
simulator = Aer.get_backend('statevector_simulator')
result = execute(circuit, backend = simulator).result()
state_vector = result.get_statevector()
print(state_vector)
```

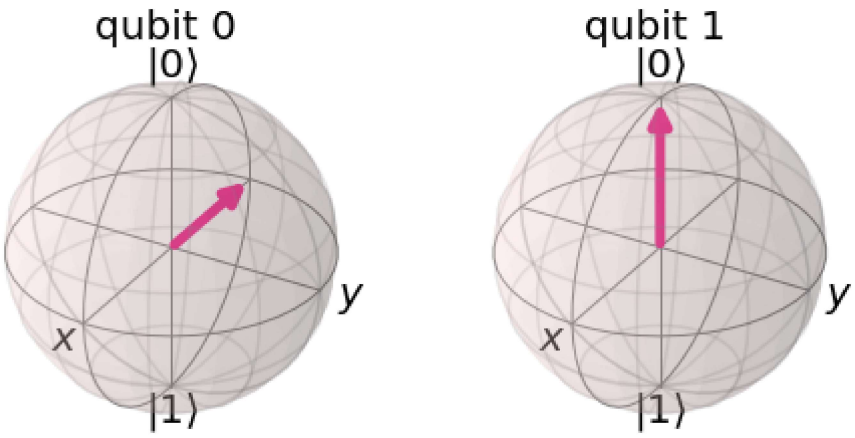
```
Statevector([ 0.70710678+0.00000000e+00j, -0.70710678-8.65956056e-17j,
              0.        +0.00000000e+00j,  0.        +0.00000000e+00j],
            dims=(2, 2))
```

Let us draw it on bloch sphere

In [25]:
```
from qiskit.tools.visualization import plot_bloch_multivector
```

In [26]:
```
1 plot_bloch_multivector(state_vector)
```

Out[26]:



Let's measure the qubits

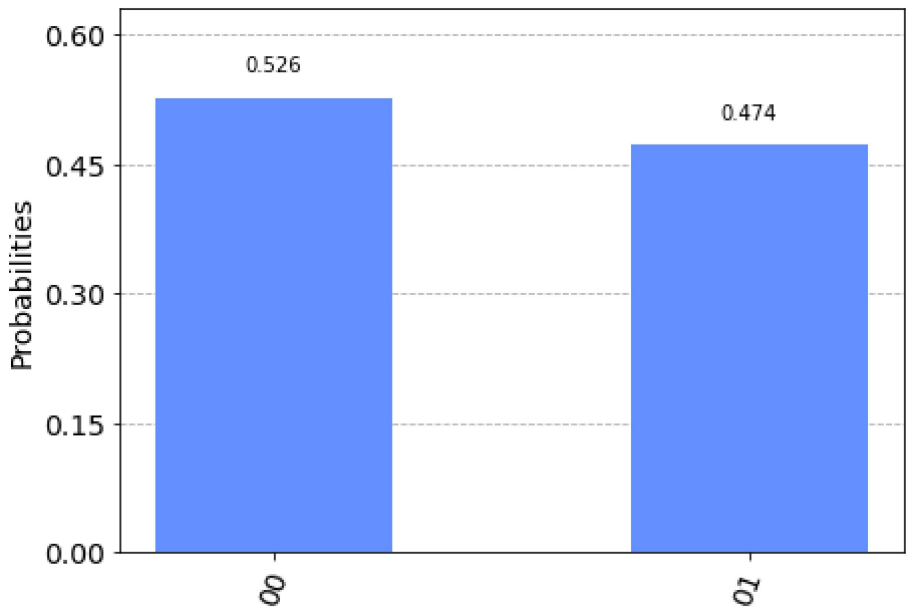
In [27]:
```
1 circuit.measure(qr,cr)
```

Out[27]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x1f42c427098>
```

Let's plot the measurement on hisogram with 1024 shots.

In [28]:
```
1 sim = Aer.get_backend('qasm_simulator')
2 result = execute(circuit, backend = sim, shots = 1024).result()
3 counts = result.get_counts()
4 from qiskit.tools.visualization import plot_histogram
5 plot_histogram(counts)
```

Out[28]:



Comparing with result from class

The new circuit applied 2 gates on the first qubit only, which flips it and then rotate it 90 degrees to the x-y plane. So the probability of the first qubit in state $|0\rangle$ and $|1\rangle$ are 50-50. Thus we have 50-50 chance of getting $|00\rangle$ and $|01\rangle$ states. (While in class we had 50-50 chance of getting $|10\rangle$ and $|11\rangle$ states)

My first circuit - IBM Quantum

https://quantu...

从 Firefox 导入 | 课程 | Coding and Basics | Physics | TA | 工具 | 生活购物tips | Gmail | YouTube | Dashboard | 英语学习 | 资源网站 | Other favorites

IBM Quantum Composer

Composer files

1 files

New file +

Name	Updated
My first circuit	2 minutes ago

File Edit Inspect View Share

Setup and run

My first circuit *Saved* A Bell state

Visualizations seed 2711

H ⊕ I T S Z T† S† P RZ |0⟩ if √X √X† Y RX RY U RXX RZZ + Add

q 0 — H

q 1

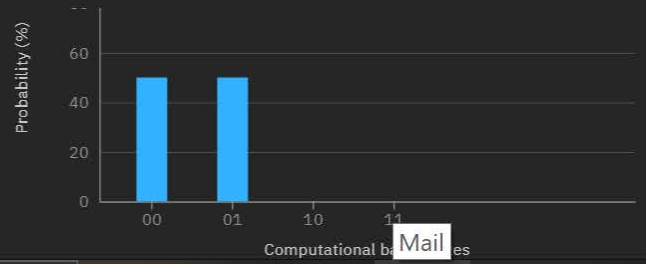
c2

OpenQASM 2.0

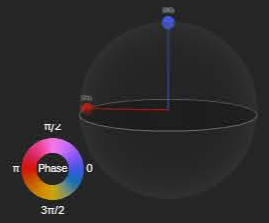
Open in Quantum Lab

```
1  OPENQASM 2.0;
2  include "qelib1.inc";
3
4  qreg q[2];
5  creg c[2];
6
7  x q[0];
8  h q[0];
```

Probabilities



Q-sphere



State Phase angle

Mail

Type here to search

51°F ENG 10:15 PM 1/4/2022