In [1]:
```python
from qiskit import *
from math import pi
import numpy as np
from qiskit.visualization import plot_bloch_multivector,plot_histogram
from qiskit_textbook.tools import array_to_latex
```

In [2]:
```python
backend_s = Aer.get_backend('statevector_simulator')
backend_u = Aer.get_backend('unitary_simulator')
```

Multiple Qubits and Entangled States

Solutions

Please note that the circuits follow the qiskit ordering of qubits, where the topmost qubit in the circuit is written in the leftmost place.

Quick Exercises 1.1

1. Write down the tensor product of the qubits:

a) $|0\rangle|1\rangle$

$|0\rangle$ = $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle$ = $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$|0\rangle|1\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ 0 \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

An interesting thing to notice, which can possibly serve as a shortcut to verifying that your result is right, is that the position of the 1 is at the index '1' when indices start from 0. The first entry of the final state corresponds to $|00\rangle$, the second to $|01\rangle$ as we see above. The following should make this observation clear:

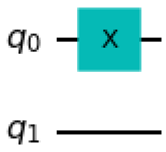
The subscripts in the following are the indices (beginning from 0) in binary: $\begin{bmatrix} 0_{00} \\ 0_{01} \\ 0_{10} \\ 0_{11} \end{bmatrix}$

So if we have the state $|00\rangle$, we just put a 1 on the index corresponding to 00, like so: $|00\rangle = \begin{bmatrix} 1_{00} \\ 0_{01} \\ 0_{10} \\ 0_{11} \end{bmatrix}$

or, to make it absolutely evident, $|00\rangle = \begin{bmatrix} 1_{|00\rangle} \\ 0_{|01\rangle} \\ 0_{|10\rangle} \\ 0_{|11\rangle} \end{bmatrix}$

In [3]:
```
qc = QuantumCircuit(2)
qc.x(0) # NOT gate to convert default 0 state to 1
qc.draw('mpl')
```

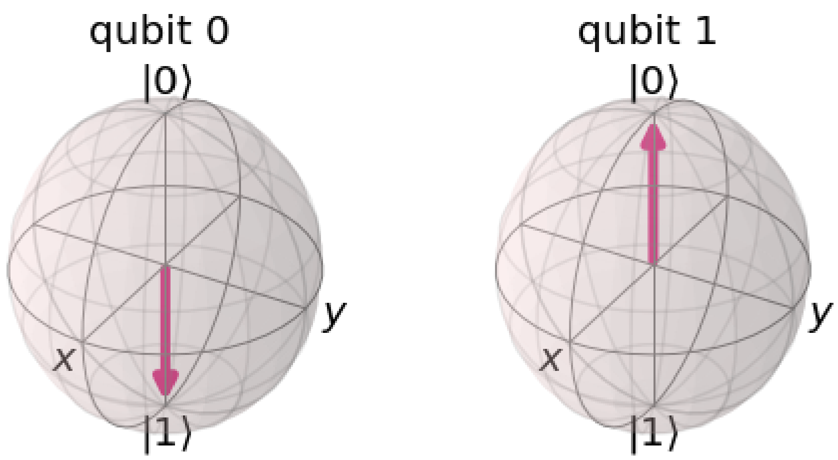
Out[3]:



In [4]:
```
result_statevector = execute(qc,backend_s).result().get_statevector()
array_to_latex(result_statevector,pretext = "\\text{Result = }")
plot_bloch_multivector(result_statevector)
```

$$\text{Result} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Out[4]:



So what do we do if we have superpositions? You may already have figured it out. Let's look at the output of Hadamard gate for a single qubit initialized to $|0\rangle$.

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1_{|0\rangle} \\ 1_{|1\rangle} \end{bmatrix} = \frac{1 \cdot |0\rangle + 1 \cdot |1\rangle}{\sqrt{2}}$$

Let us look at the remaining exercises and arrive at our solutions both ways: Regular matrix tensor product and our shortcut. Finally we can look at the output from the statevector simulator.

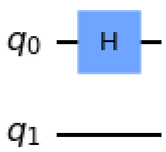
b) $|0\rangle|+\rangle$

$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|+\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$|0\rangle|+\rangle = |0\rangle \otimes |+\rangle = |0\rangle \otimes \frac{(|0\rangle+|1\rangle)}{\sqrt{2}} = \frac{(|00\rangle+|01\rangle)}{\sqrt{2}} = \begin{bmatrix} 1\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ 0\begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1_{|00\rangle} \\ 1_{|01\rangle} \\ 0_{|10\rangle} \\ 0_{|11\rangle} \end{bmatrix}$$

In [5]:
```
qc = QuantumCircuit(2)
qc.h(0)
qc.draw('mpl')
```

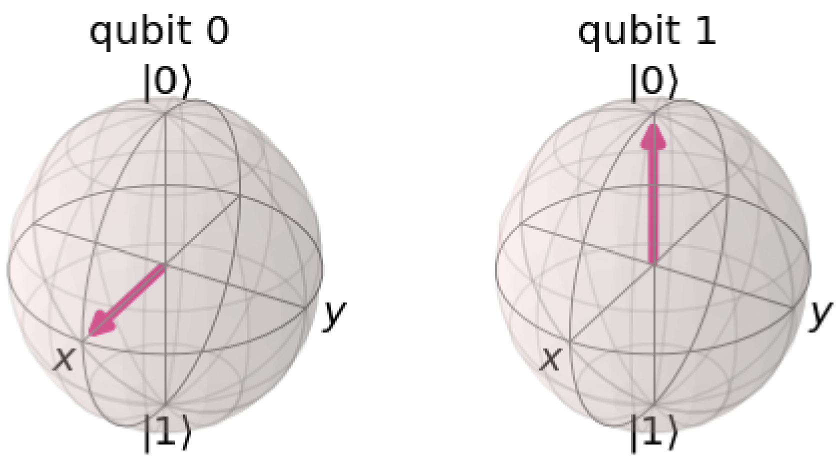
Out[5]:



In [6]:
```
result_statevector = execute(qc,backend_s).result().get_statevector()
array_to_latex(result_statevector,pretext = "\\text{Result = }")
plot_bloch_multivector(result_statevector)
```

$$\text{Result} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix}$$

Out[6]:



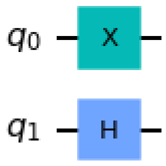
c) $|+\rangle|1\rangle$

$|+\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$|+\rangle|1\rangle = |+\rangle \otimes |1\rangle = \frac{(|0\rangle+|1\rangle)}{\sqrt{2}} \otimes |1\rangle = \frac{(|01\rangle+|11\rangle)}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1\begin{bmatrix}0\\1\end{bmatrix} \\ 1\begin{bmatrix}0\\1\end{bmatrix}\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}0\\1\\0\\1\end{bmatrix} = \frac{1}{\sqrt{2}}$$

In [7]:
```
qc = QuantumCircuit(2)
qc.x(0)
qc.h(1)
qc.draw('mpl')
```

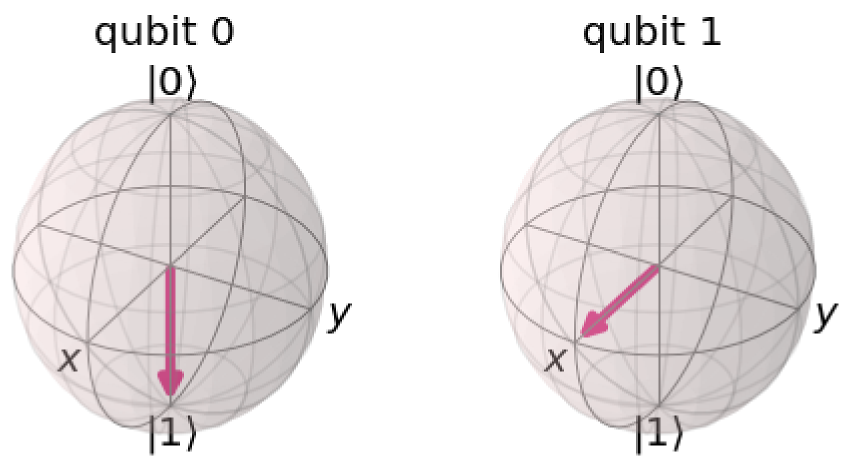
Out[7]:



In [8]:
```
result_statevector = execute(qc,backend_s).result().get_statevector()
array_to_latex(result_statevector,pretext = "\\text{Result = }")
plot_bloch_multivector(result_statevector)
```

$$\text{Result} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Out[8]:



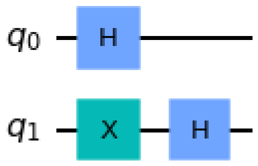
d) $|-\rangle|+\rangle$

$|-\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$ and $|+\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$

$$|-\rangle|+\rangle = |-\rangle \otimes |+\rangle = \frac{(|0\rangle-|1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle+|1\rangle)}{\sqrt{2}} = \frac{(|00\rangle+|01\rangle-|10\rangle-|11\rangle)}{2} = \frac{1}{2}\begin{bmatrix} 1\begin{bmatrix}1\\1\end{bmatrix} \\ -1\begin{bmatrix}1\\1\end{bmatrix}\end{bmatrix} =$$

In [9]:
```
qc = QuantumCircuit(2)
qc.h(0)
qc.x(1)
qc.h(1)
qc.draw('mpl')
```

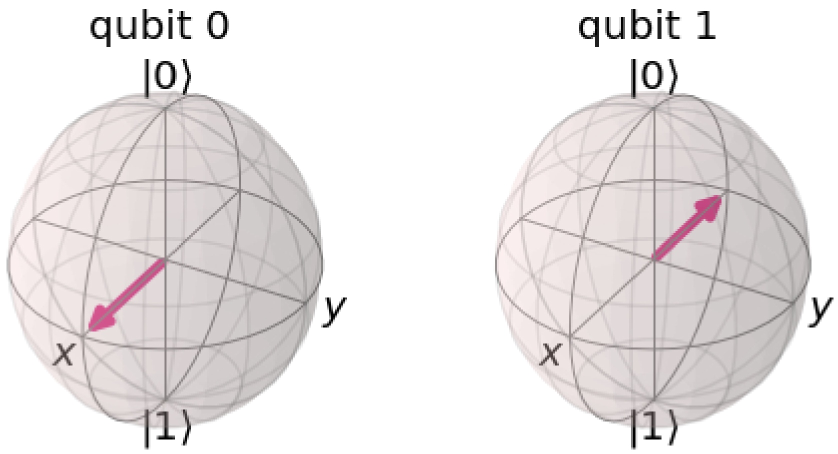
Out[9]:



In [10]:
```
result_statevector = execute(qc,backend_s).result().get_statevector()
array_to_latex(result_statevector,pretext = "\\text{Result = }")
plot_bloch_multivector(result_statevector)
```

$$\text{Result} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

Out[10]:



2. Write the state $|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{i}{\sqrt{2}}|01\rangle$ as two separate qubits.

We can see that the first qubit(rightmost) is 0 in both the terms of $|\psi\rangle$, hence that can be "taken common" from the above and written as the tensor product with the other qubit in the following way:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{i}{\sqrt{2}}|01\rangle = |0\rangle \otimes (\frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle)$$

Quick Exercises 2.1

1. Calculate the single qubit unitary (U) created by the sequence of gates: $U = XZH$. Use Qiskit's unitary simulator to check your results.

The question is pretty straightforward. For reference, the following are the gates' matrix representations:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

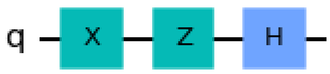
Simple multiplication of all three gives us:

In [11]:

```
qc = QuantumCircuit(1)
qc.x(0)
qc.z(0)
qc.h(0)

qc.draw('mpl')
```

Out[11]:



In [12]:

```
result_unitary = execute(qc,backend_u).result().get_unitary()
array_to_latex(result_unitary,pretext = "\\text{U = XZH = }")
```

$$\text{U = XZH = } \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

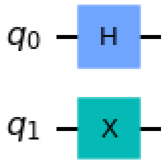
2. Try changing the gates in the circuit above. Calculate their tensor product, and then check your answer using the unitary simulator.

Circuit:

In [13]:
```python
qc = QuantumCircuit(2)
qc.h(0)
qc.x(1)

qc.draw('mpl')
```

Out[13]:



In [14]:
```python
array_to_latex(execute(qc,backend_u).result().get_unitary())
```

$$\begin{bmatrix} 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \end{bmatrix}$$

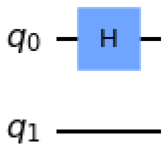
which is equal to $\begin{bmatrix} 0 & H \\ H & 0 \end{bmatrix} = X \otimes H$

Let us remove the X gate from q_1

In [15]:
```python
qc = QuantumCircuit(2)
qc.h(0)

qc.draw('mpl')
```

Out[15]:



In [16]:
```python
array_to_latex(execute(qc,backend_u).result().get_unitary())
```

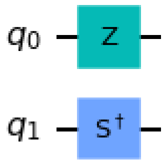
$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

which is equal to $\begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix} = I \otimes H$

As a last alteration in this notebook, let's try new gates:

In [17]:
```
qc = QuantumCircuit(2)
qc.sdg(1)
qc.z(0)
qc.draw('mpl')
```

Out[17]:



In [18]:
```
array_to_latex(execute(qc,backend_u).result().get_unitary())
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -i & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

which is equal to $\begin{bmatrix} Z & 0 \\ 0 & -i \cdot Z \end{bmatrix} = S^\dagger \otimes Z$

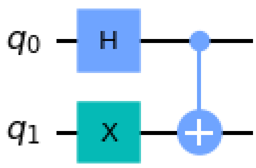
Quick Exercises 3.3

1. Create a quantum circuit that produces the Bell state: $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$. Use the statevector simulator to verify your result.

In [19]:
```
qc = QuantumCircuit(2)
qc.h(0)
qc.x(1)
qc.cx(0,1)

qc.draw('mpl')
```

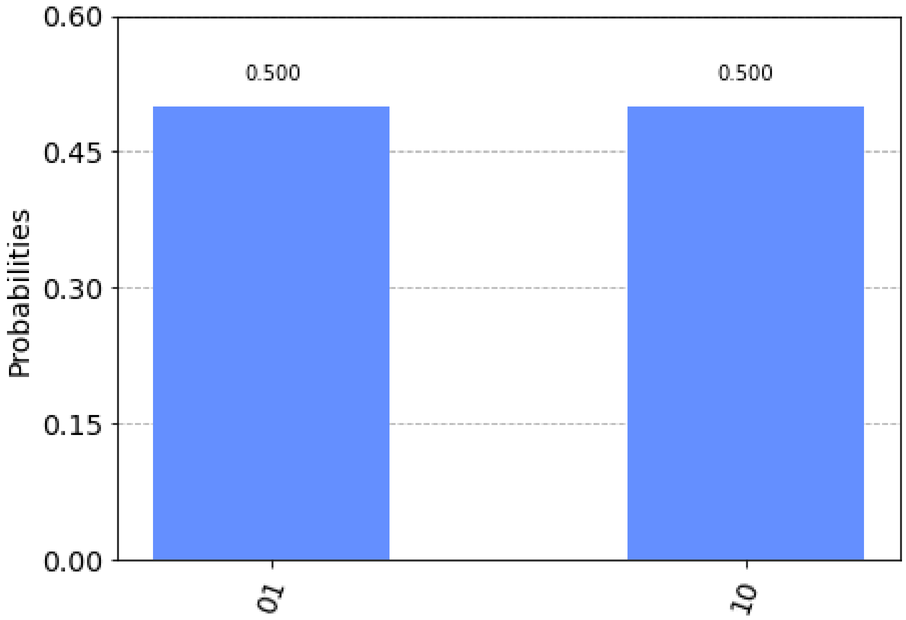
Out[19]:



In [20]:
```python
entangled = execute(qc,backend_s).result()
entangled_statevector = entangled.get_statevector()
entangled_counts = entangled.get_counts()
array_to_latex(entangled_statevector, pretext = "\\text{Statevector = }")
plot_histogram(entangled_counts)
```

$$\text{Statevector} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

Out[20]:



In [21]:
```python
import qiskit
qiskit.__qiskit_version__
```

Out[21]:
```
{'qiskit-terra': '0.16.1',
 'qiskit-aer': '0.7.1',
 'qiskit-ignis': '0.5.1',
 'qiskit-ibmq-provider': '0.11.1',
 'qiskit-aqua': '0.8.1',
 'qiskit': '0.23.1'}
```

In [ ]:
```python
```