In [83]:

```
# Continuation of CNOT ID1 is CNOT gate reverse, here we get same output
# refer to http://localhost:8888/notebooks/CNOT%20ID1.ipynb
# https://www.youtube.com/watch?v=uNrPJ3_Mttc
from qiskit import*
from qiskit.visualization import visualize_transition, plot_histogram, plot_bloch_mult
```

In [84]:

```
#import qiskit_textbook and display the unitary matrix
from qiskit.quantum_info import Statevector
from qiskit.visualization import array_to_latex
```

In [85]:

```
# Create a quantum Circuit with 1 qubits
qc= QuantumCircuit(1)
state = Statevector.from_instruction(qc)
```

In [86]:

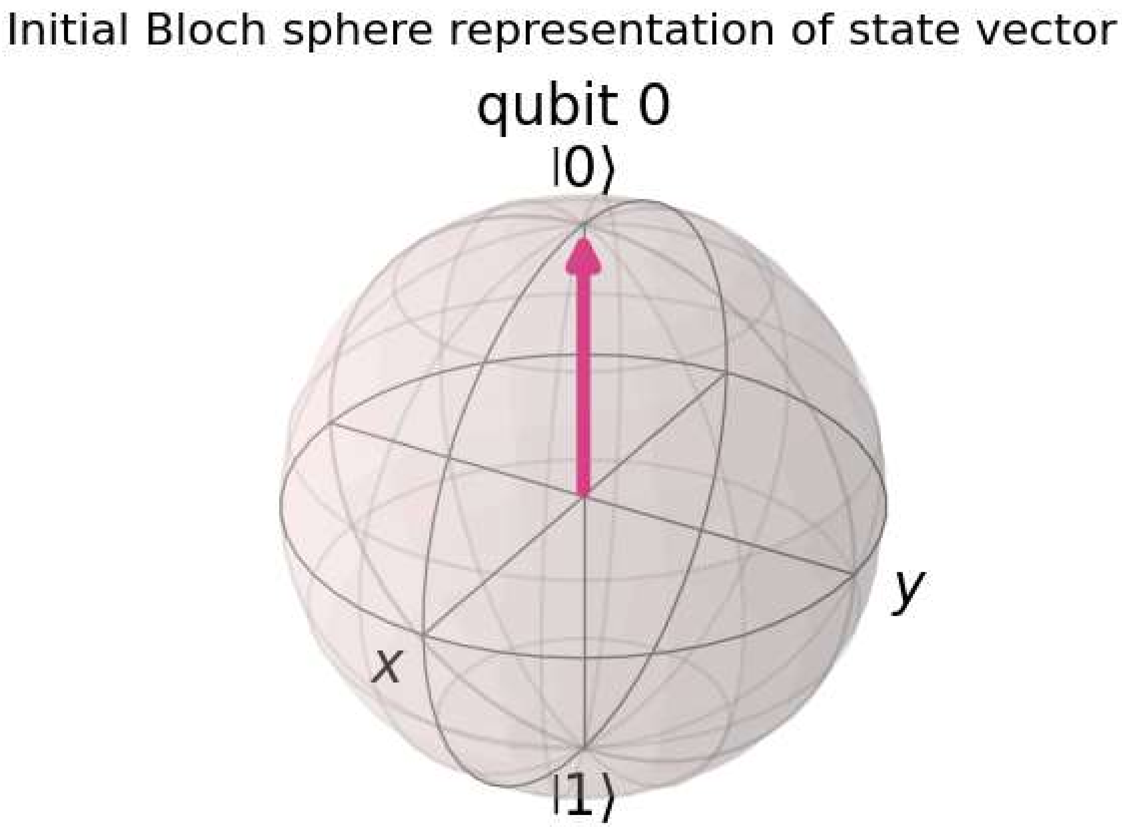
```
#Draw the circuit
#qc.draw()
qc.draw('mpl')
```

Out[86]:

In [87]:

```
#draw the initial bloch sphere
state.draw('Bloch', title = 'Initial Bloch sphere representation of state vector')
```

Out[87]:


Initial Bloch sphere representation of state vector

In [88]:

```
# draw the latec
state.draw('latex', prefix= '\\text{Statevector} \\psi\\rangle = ')
```

Out[88]:

$$\text{Statevector}\psi\rangle = |0\rangle$$

In [89]:

```
# Observe above initial state before applying the gate
#Apply the X/Y/Z gates in the below and extract the output in different forms like 'la
qc.y(0)
state = Statevector.from_instruction(qc)
state.draw('latex', prefix= '\\text{Statevector} |\\psi\\rangle = ')
```

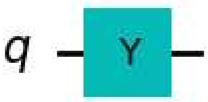
Out[89]:

$$\text{Statevector}|\psi\rangle = i|1\rangle$$

In [90]:

```
#Draw the circuit
#qc.draw()
qc.draw('mpl')
```

Out[90]:



In [91]:

```
state.draw('text', prefix= '\\text{Statevector} |\\psi\\rangle = ')
```

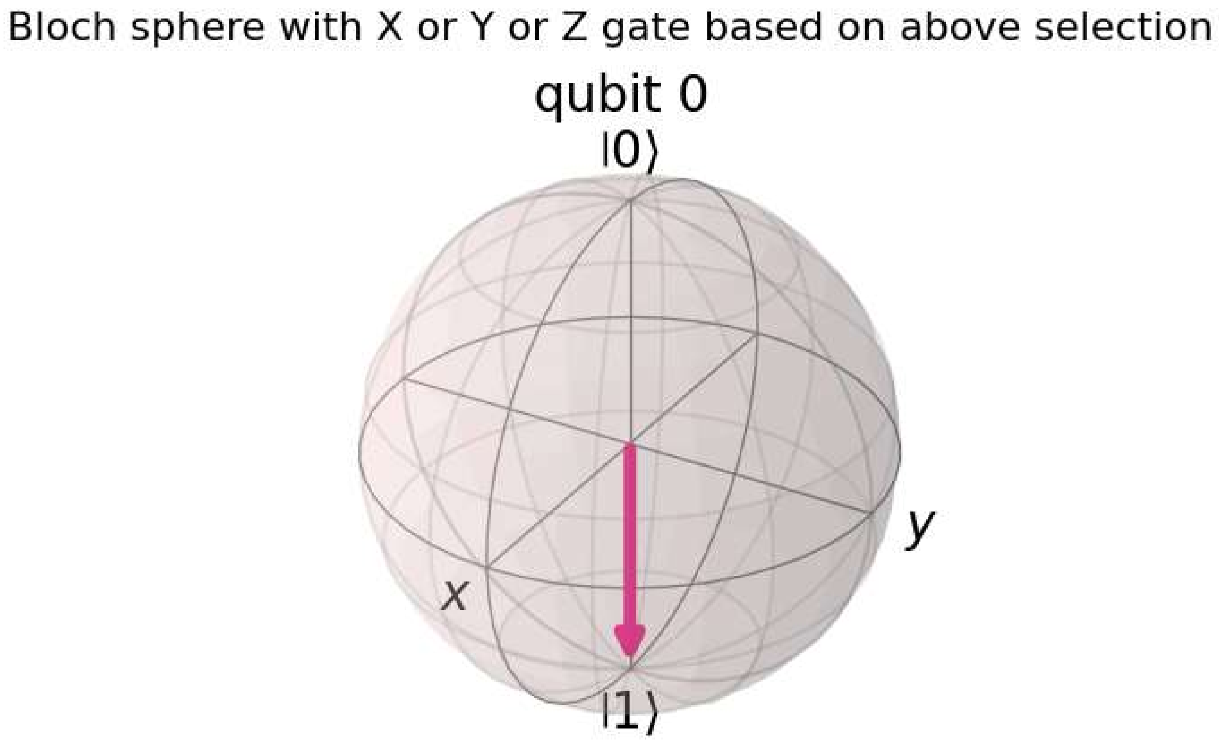
Out[91]:

```
\text{Statevector} |\psi\rangle = [0.+0.j,0.+1.j]
```

In [92]:

```
#draw the initial bloch sphere
state.draw('Bloch', title = 'Bloch sphere with X or Y or Z gate based on above selecti
```

Out[92]:



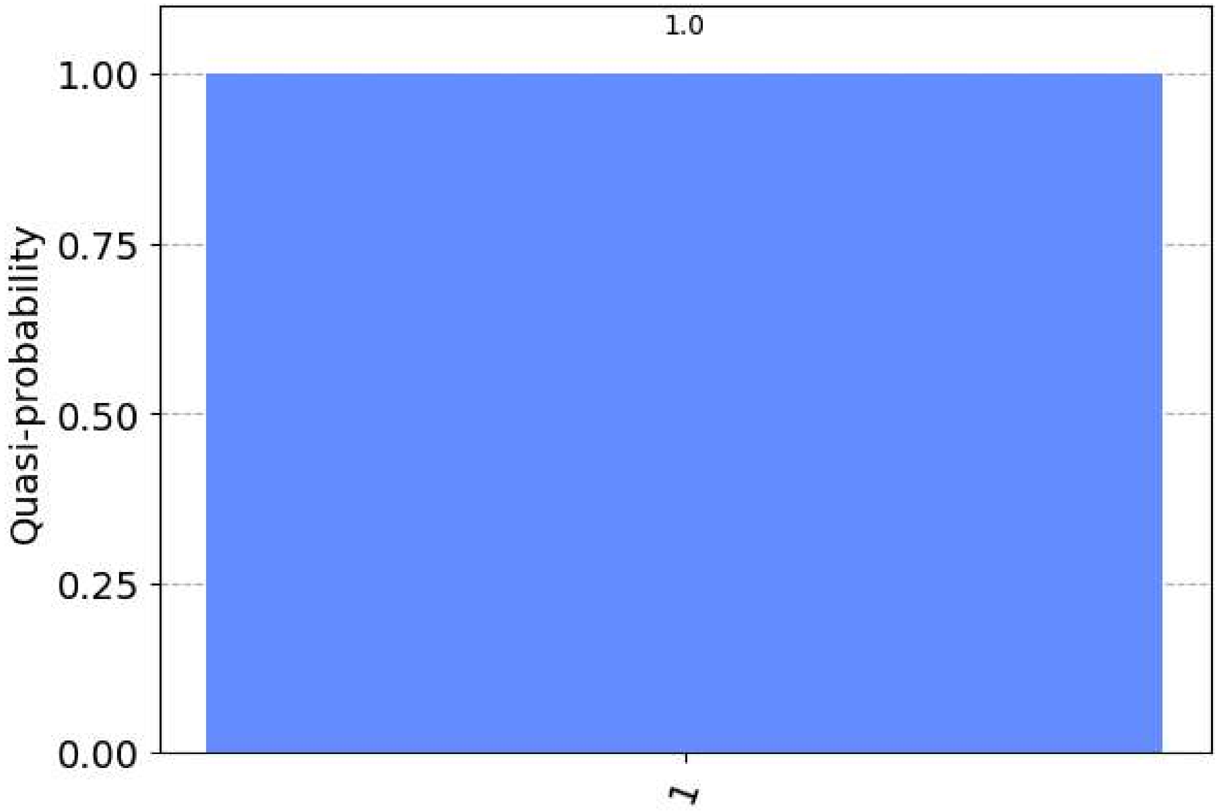
In [93]:

```
#simulator = Aer.get_backend('qasm_simulator')
#result = execute(qc,backend=simulator, shots=1).result()
#counts = out.get_counts()
#print(counts)
```

In [94]:

```
plot_histogram([counts])
```

Out[94]:



```
In [ ]:
```