

```
1 # Import necessary modules (compatible with Qiskit 2.x and Aer 0.17.x)
2 from qiskit import QuantumCircuit, transpile
3 from qiskit_aer import AerSimulator
4 from qiskit.visualization import plot_histogram
5 import matplotlib.pyplot as plt
6 # Create a quantum circuit with 1 qubit and 1 classical bit for measurement
7 qc = QuantumCircuit(1, 1)
8 # Apply Hadamard gate to put the qubit in superposition (equal chance of |0> or |1>)
9 qc.h(0)
10 # Measure the qubit
11 qc.measure(0, 0)
12 # Set up the simulator
13 simulator = AerSimulator()
14 # Transpile the circuit for the simulator (optimizes for the backend)
15 qc_compiled = transpile(qc, simulator)
16 # Run the circuit 1024 times (shots) to see probabilistic outcomes
17 job = simulator.run(qc_compiled, shots=1024)
18 result = job.result()
19 counts = result.get_counts()
20 # Plot the results as a histogram
21 plot_histogram(counts)
22 plt.show()
23 # Print the counts for reference
24 print("Measurement outcomes:", counts)
```