QISKIT CHEAT SHEET

Step 1: Creating Quantum Circuits	
Creating a quantum circuit with **1 qubit**	`qc = QuantumCircuit(1)`
Creating a quantum circuit with **n qubits**	`qc = QuantumCircuit(n)`
Creating a quantum circuit with **n qubits and m classical bits**	`qc = QuantumCircuit(n, m)`

Step 2: Quantum Gates	
X gate on qubit 0	`qc.x(0)`
X gate on qubit a (indexed by 0)	`qc.x(a)`
Z gate on qubit a (indexed by 0)	`qc.z(a)`
H gate on qubit a (indexed by 0)	`qc.h(a)`
CX gate with qubit a as the control and qubit b as the target	`qc.cx(a, b)`

Step 2: Measurement	
Measuring the state of qubit 0 and storing the result in classical bit 0	`qc.measure(0, 0)`
Measuring the states of qubits 0 and 1 and storing the results in classical bits 0 and 1	`qc.measure([0, 1], [0, 1]`
Measuring the states of qubits 0, 1, 2 and storing the results in classical bits 0, 1, 2	`qc.measure([0, 1, 2], [0, 1, 2]`
Measuring the states of qubits 0 on and storing the results in classical bits 0 on	`qc.measure([0, 1, 2, ...], [0, 1, 2, ...]`

Step 3: Visualizing and Running Circuits	
Draw the circuit	`qc.draw()`
Visualize the rotation on the Bloch sphere. NOTE: This works for 1 qubit circuits only.	`visualize_transition(qc, trace = True)`

Simulate using the QASM simulator. You are not responsible for knowing this code, you just need to know how to interpret the results.	`# Create a QASM simulator`
	`backend = Aer.get_backend('qasm_simulator')`
	
	`# Simulate your specific circuit, make sure`
	`# to use the correct variable name (qc here)`
	`job = execute(qc, backend=backend, shots=1024)`
	
	`# Get the counts from the simulation result`
	`result = job.result()`
	`counts = result.get_counts()`
	
	`# Plot the results as a histogram`
	`plot_histogram(counts)`