

HELLO QUBIT

SOURCE CODE:

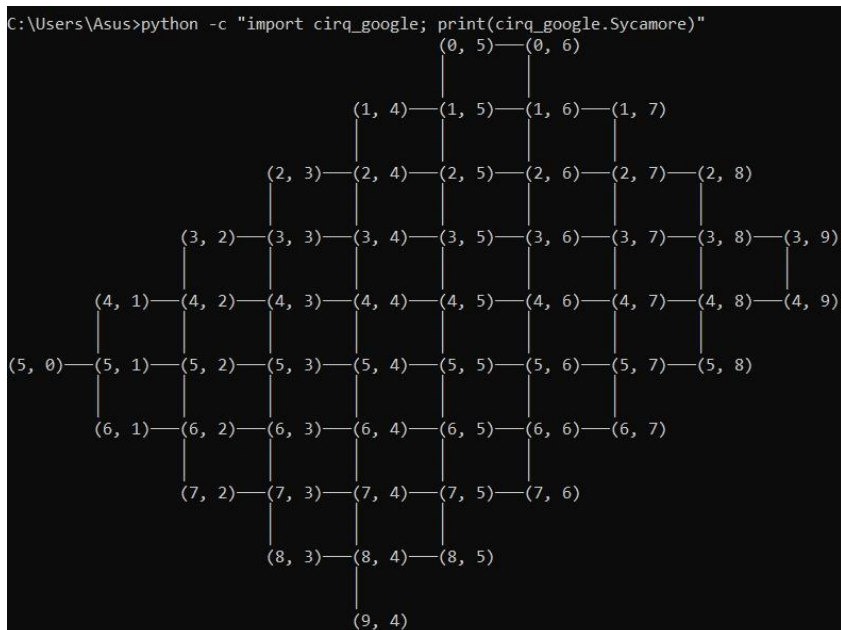
```
try:
    import cirq
except ImportError:
    print("installing cirq...")
    !pip install --quiet cirq
    import cirq
    print("installed cirq.")

# Pick a qubit.
qubit = cirq.GridQubit(0, 0)

# Create a circuit
circuit = cirq.Circuit(
    cirq.X(qubit)**0.5, # Square root of NOT.
    cirq.measure(qubit, key='m') # Measurement.
)
print("Circuit:")
print(circuit)

# Simulate the circuit several times.
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=20)
print("Results:")
print(result)
```

OUTPUT :



SOURCE CODE:

```
import cirq
a = cirq.NamedQubit('s')
b = cirq.NamedQubit('t')
a, b, c = cirq.LineQubit.range(3)
d = cirq.LineQubit(6)
e = cirq.GridQubit(4, 5)
print(a)
print(b)
print(c)
print(d)
print(e)
```

OUTPUT :

```
q(0)
q(1)
q(2)
q(6)
q(4, 5)
```

SOURCE CODE:

```
import cirq
a= cirq.Circuit()
b= cirq.LineQubit.range(4)
a.append(cirq.H(b[0]))
a.append(cirq.H(b[1]))
a.append(cirq.H(b[2]))
a.append(cirq.H(b[3]))

print(a)
```

OUTPUT :

```
0: ——H——
1: ——H——
2: ——H——
3: ——H——
```

SOURCE CODE:

```
import cirq
a = cirq.NamedQubit('a')
b = cirq.NamedQubit('s')
c = cirq.NamedQubit('c')
```

```

print(cirq.H(b))
print(cirq.CNOT(b, c))
print(cirq.CNOT(a, b))
print(cirq.H(a))
print(cirq.measure(a,b))

```

OUTPUT :

```

H(s)
CNOT(s, c)
CNOT(a, s)
H(a)
cirq.MeasurementGate(2, cirq.MeasurementKey(name='a,s'), ())(a, s)

```

SOURCE CODE:

```

import cirq
cnot= cirq.CNOT
pauli= cirq.Z
sqrt1= cirq.X**0.5
sqrt2= cirq.YPowGate(exponent=0.25)
a, b= cirq.LineQubit.range(2)
swap= cirq.SQRT_ISWAP(a, b)
c= cnot(a, b)
d= pauli(a)
e= sqrt1(b)
f= sqrt2(a)
print(c)
print(d)
print(e)
print(f)
print(swap)

```

OUTPUT :

```

CNOT(q(0), q(1))
Z(q(0))
X**0.5(q(1))
Y**0.25(q(0))
ISWAP**0.5(q(0), q(1))

```

SOURCE CODE:

```

import cirq
a = cirq.GridQubit(0, 1)
b = cirq.Circuit(
    cirq.X(a)**0.5,
    cirq.measure(a, key='m')
)
print("Circuit:")
print(b)

s = cirq.Simulator()
r = s.run(b, repetitions=15)

```

```
print("Results:")
print(r)
```

OUTPUT :

```
Circuit:
(0, 1): —X^0.5—M('m')—
Results:
m=011000111010111
```

SOURCE CODE:

```
print(cirq.Circuit(cirq.SWAP(q, q + 1) for q in cirq.LineQubit.range(4)))
```

OUTPUT :

```
0: —x————
    |
1: —x—x—
    |
2: —x—x—
    |
3: —x—x—
    |
4: —x—
```

SOURCE CODE:

```
import cirq
a= cirq.CZ(qubits[0], qubits[1])
b= cirq.X(qubits[2])
d= cirq.Moment(b,a)

print(d)
```

OUTPUT :

```
  | 0 1 2
--|-----
0 | @-@ X
  |
```