トポロジカル量子計算　勉強会　第1回

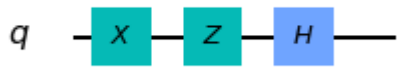
1ビットの量子回路をつくる

```
# Useful additional packages
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import math
```

```
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister, execute
from qiskit.tools.visualization import circuit_drawer
from qiskit import BasicAer
```

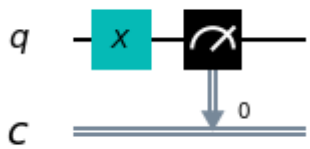
ゲートを並べて量子回路を作る

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



測定を含む回路１

```
qc = QuantumCircuit(1,1)
qc.x(0)
qc.measure(0,0)
qc.draw()
```

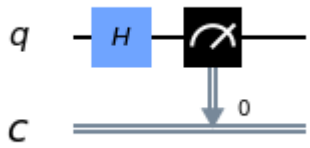



```
backend = BasicAer.get_backend('qasm_simulator')
job = execute(qc, backend, shots=1000)
job.result().get_counts(qc)
```

```
{'1': 1000}
```

測定を含む回路２

```
qc = QuantumCircuit(1,1)
qc.h(0)
qc.measure(0,0)
qc.draw()
```

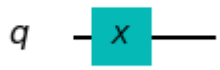



```
backend = BasicAer.get_backend('qasm_simulator')
job = execute(qc, backend, shots=1000)
job.result().get_counts(qc)
```

```
{'0': 521, '1': 479}
```

作成した回路のユニタリ行列を確認する１

```
qc = QuantumCircuit(1)
qc.x(0)
qc.draw()
```

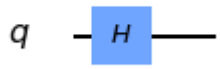



```
backend = BasicAer.get_backend('unitary_simulator')
job = execute(qc, backend)
print(job.result().get_unitary(qc, decimals=3))
```

```
[[0.+0.j 1.+0.j]
 [1.+0.j 0.+0.j]]
```

作成した回路のユニタリ行列を確認する２

```
qc = QuantumCircuit(1)
qc.h(0)
qc.draw()
```



```
backend = BasicAer.get_backend('unitary_simulator')
job = execute(qc, backend)
print(job.result().get_unitary(qc, decimals=3))
```

```
[[ 0.707+0.j  0.707+0.j]
 [ 0.707+0.j -0.707+0.j]]
```

作成した回路のユニタリ行列を確認する２

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

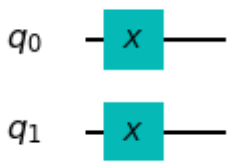

```
backend = BasicAer.get_backend('unitary_simulator')
job = execute(qc, backend)
print(job.result().get_unitary(qc, decimals=3))
```

```
[[ 1.+0.j  0.+0.j]
 [ 0.+0.j -1.+0.j]]
```

2ビットの量子回路を作る

2ビットの例１

```
qc = QuantumCircuit(2)
qc.x(0)
qc.x(1)
qc.draw()
```

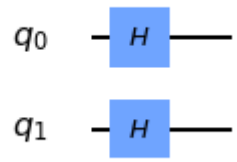



```
backend = BasicAer.get_backend('unitary_simulator')
job = execute(qc, backend)
print(job.result().get_unitary(qc, decimals=3))
```

```
[[0.+0.j 0.+0.j 0.+0.j 1.+0.j]
 [0.+0.j 0.+0.j 1.+0.j 0.+0.j]
 [0.+0.j 1.+0.j 0.+0.j 0.+0.j]
 [1.+0.j 0.+0.j 0.+0.j 0.+0.j]]
```

2ビットの例2

```
qc = QuantumCircuit(2)
qc.h(0)
qc.h(1)
qc.draw()
```

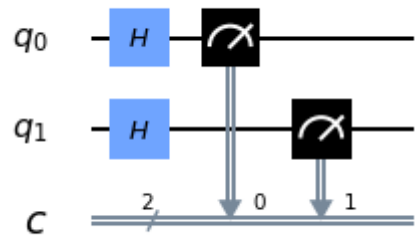



```
backend = BasicAer.get_backend('unitary_simulator')
job = execute(qc, backend)
print(job.result().get_unitary(qc, decimals=3))
```

```
[[ 0.5+0.j  0.5+0.j  0.5+0.j  0.5+0.j]
 [ 0.5+0.j -0.5+0.j  0.5+0.j -0.5+0.j]
 [ 0.5+0.j  0.5+0.j -0.5+0.j -0.5+0.j]
 [ 0.5+0.j -0.5+0.j -0.5+0.j  0.5+0.j]]
```

2ビットの回路の例１　（エンタングルなし）

```
qc = QuantumCircuit(2,2)
qc.h(0)
qc.h(1)
qc.measure(0,0)
qc.measure(1,1)
qc.draw()
```

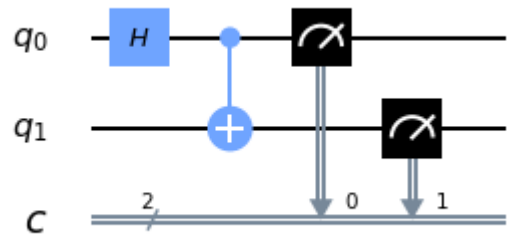



```
backend = BasicAer.get_backend('qasm_simulator')
job = execute(qc, backend, shots=1000)
job.result().get_counts(qc)
```

```
{'11': 247, '10': 265, '01': 244, '00': 244}
```

2ビットの回路の例２　（エンタングルあり）

```
qc = QuantumCircuit(2,2)
qc.h(0)
qc.cx(0,1)
qc.measure(0,0)
qc.measure(1,1)
qc.draw()
```




```
backend = BasicAer.get_backend('qasm_simulator')
job = execute(qc, backend, shots=1000)
job.result().get_counts(qc)
```

```
{'11': 495, '00': 505}
```

```
```