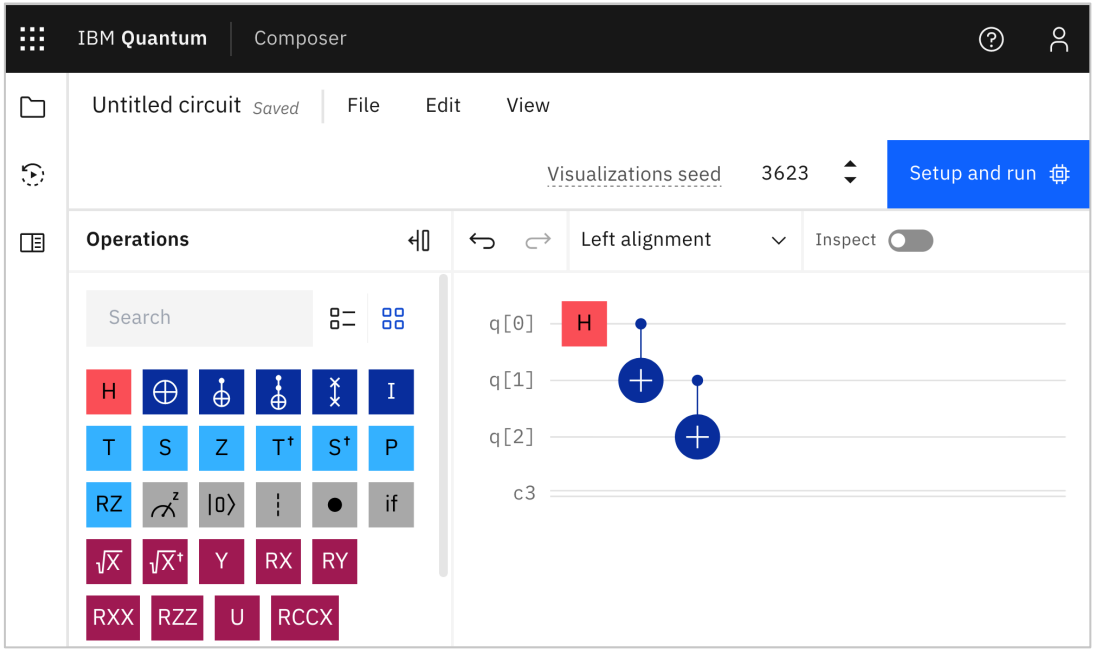
Kawasaki Quantum Summer Camp

量子ゲート基礎
IBM Quantum Composer



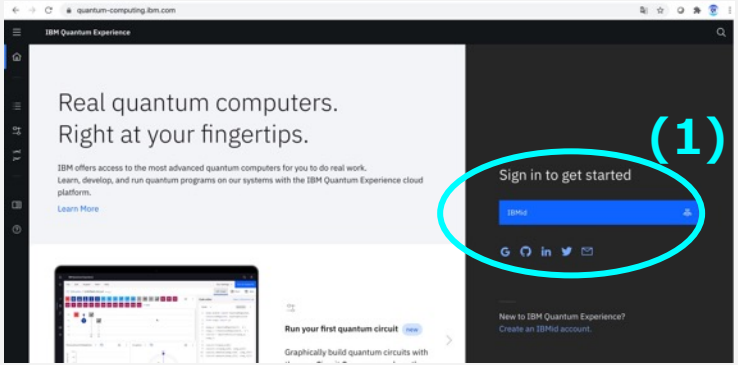
沼田 祈史
Kifumi Numata
IBM Quantum

川崎市 KAWASAKI CITY

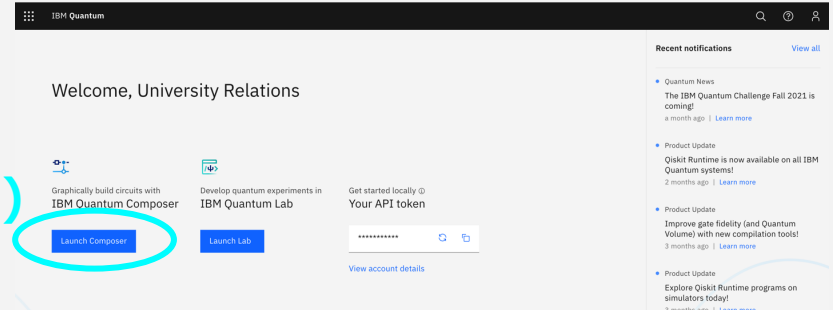
Qiskit

ハンズオン：いっしょにやってみましょう！

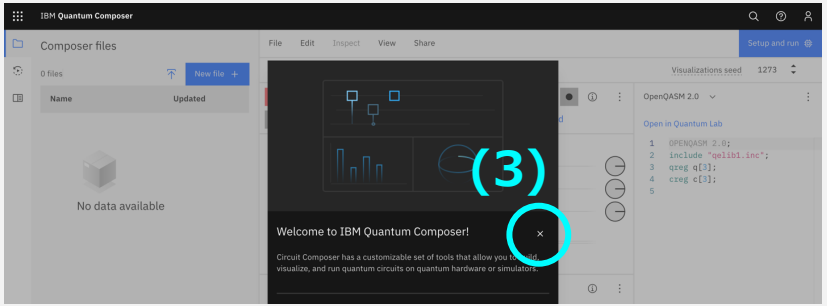
(1) IBM Quantum にログインします。URL: https://quantum-computing.ibm.com/



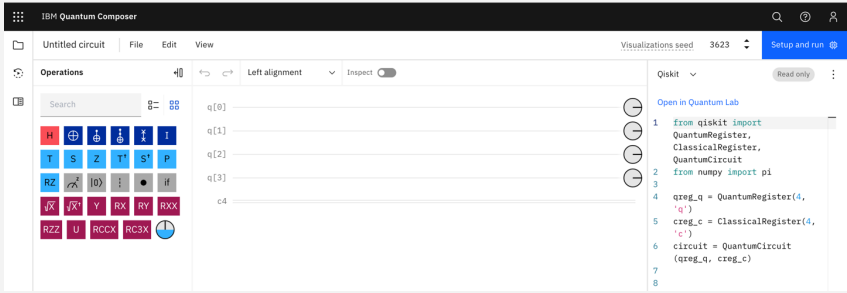
(2) 左の青アイコン「Launch Composer」をクリック。



(3) ポップアップウィンドウは「x」をクリックして、閉じます。

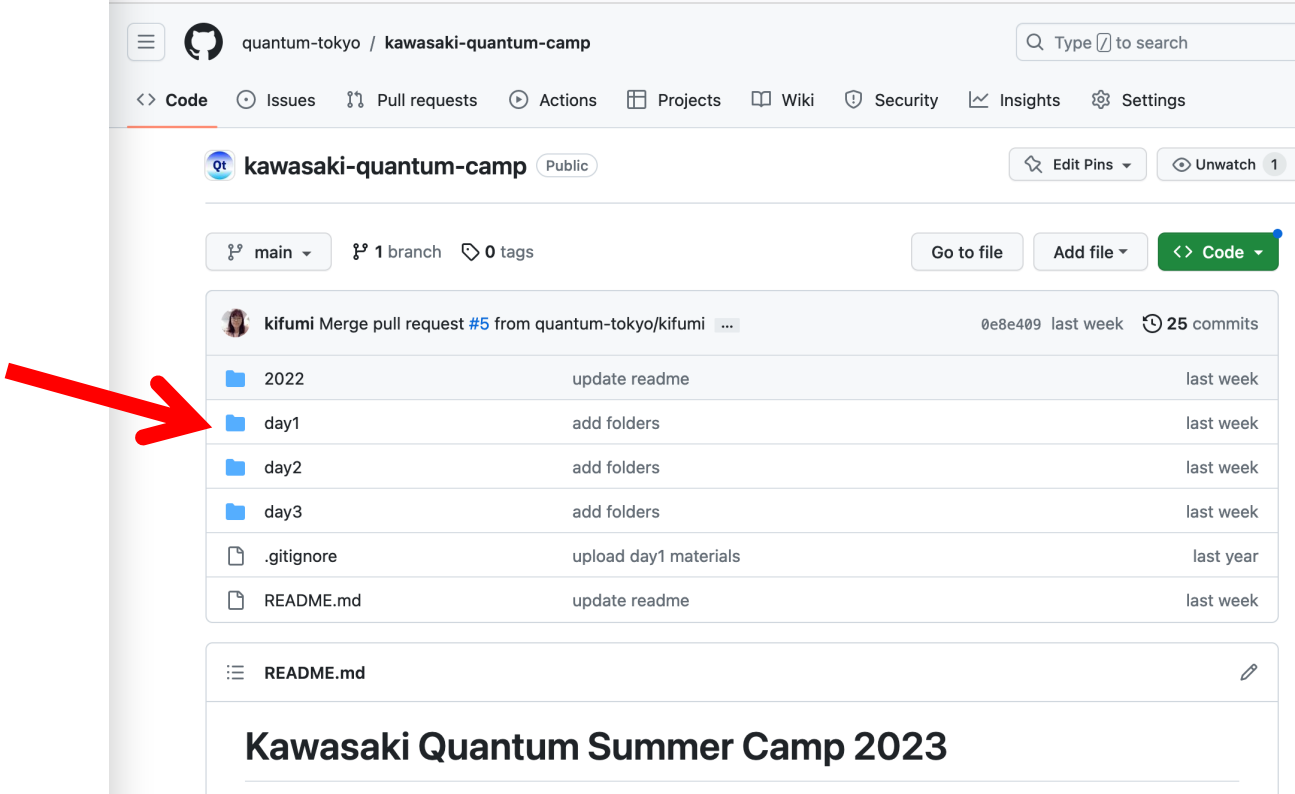


(4) この画面になったら準備完了です。

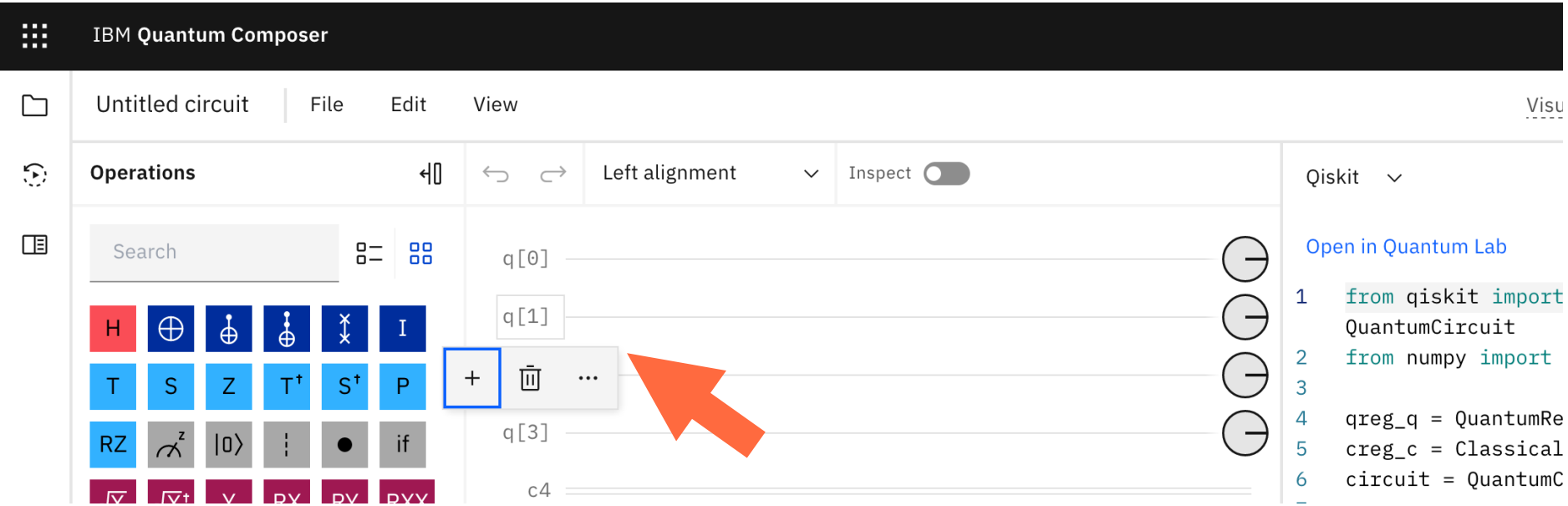


ハンズオンの資料

URL: ibm.biz/kwskgit



1量子ビット回路



マウスでq[1]をクリックするとゴミ箱マークが出てくるので、クリックして消します。

q[0]だけにして、1量子ビット回路の準備をします。

1量子ビット回路

量子ゲートをマウスでドラッグ&ドロップして、
量子回路を作ります。
右側には、Qiskitのコードが自動生成されます。

1. Xゲート(NOTゲート)

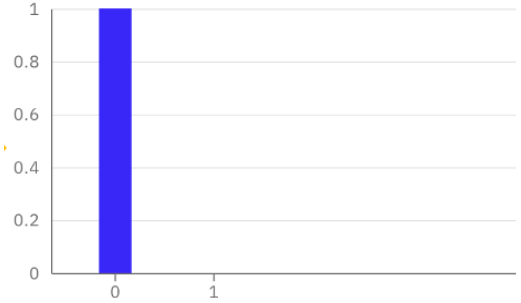
図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

1-1)

1-2)

1-3)

初期状態は|0>



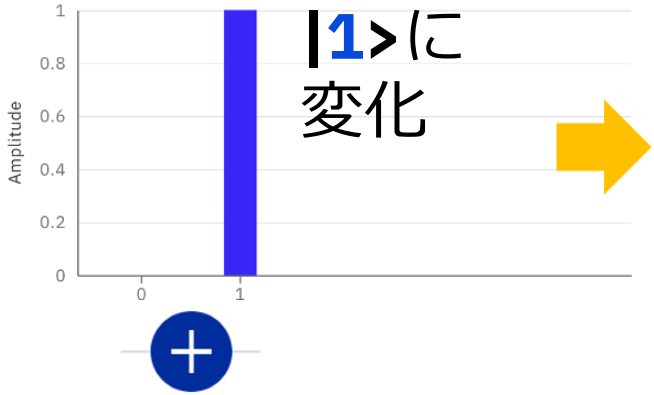
棒グラフ (Statevector 表示) は
量子ビットの状態

$$\alpha \times |0\rangle + \beta \times |1\rangle$$

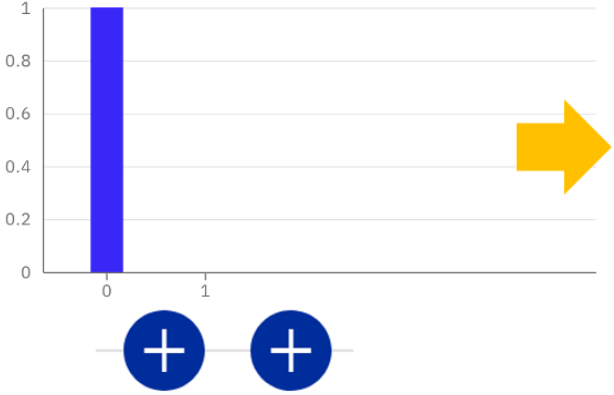
の α, β (確率振幅)です。

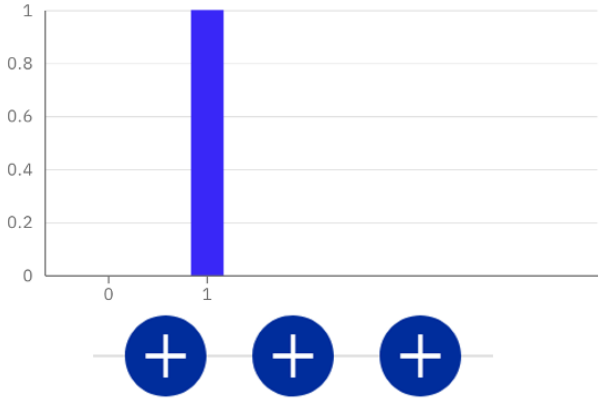
$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

$$|1\rangle = 0 \times |0\rangle + 1 \times |1\rangle$$

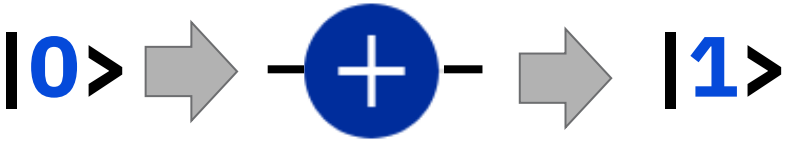


|1>に
変化

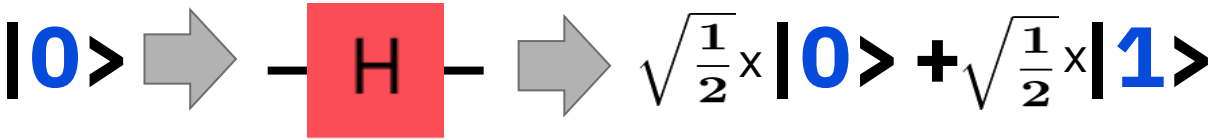




量子コンピューターの計算方法

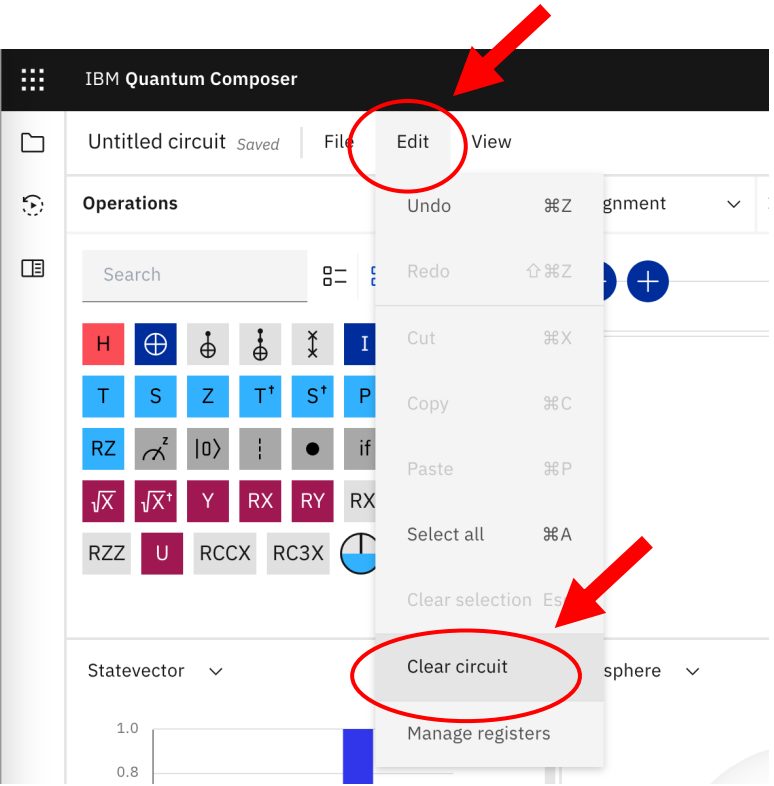


ノット（反転）ゲート

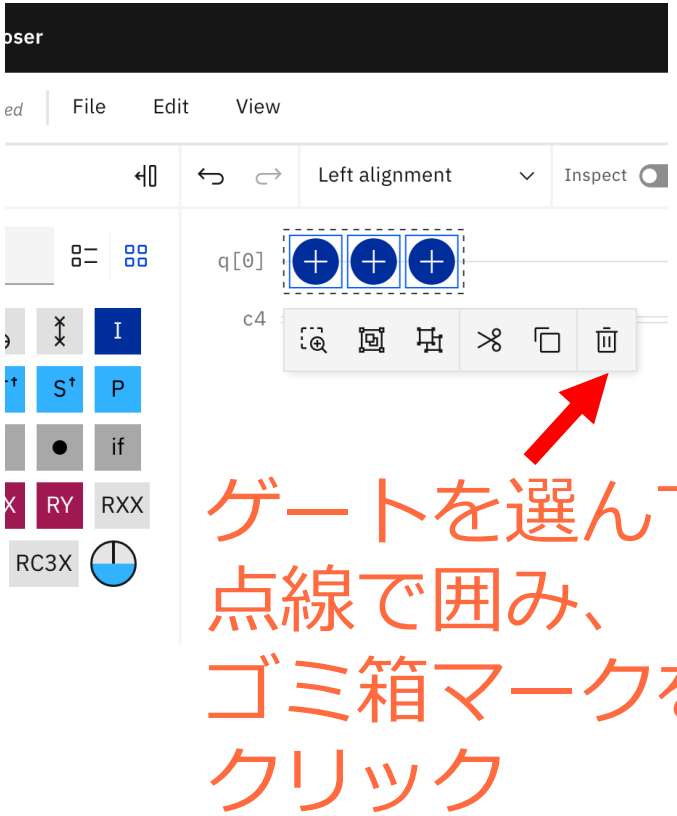


重ね合わせをつくる

置いたゲートを取り除く



または

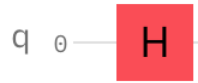


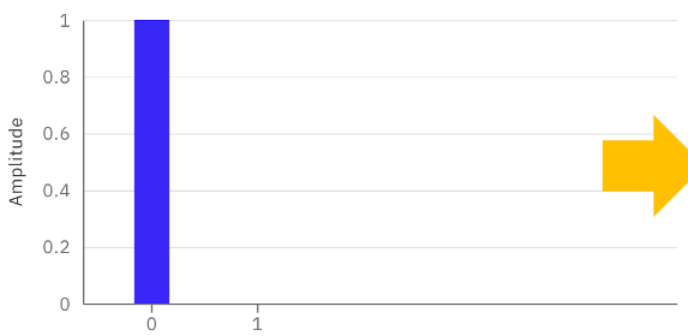
ゲートを選んで
点線で囲み、
ゴミ箱マークを
クリック

2. Hゲート

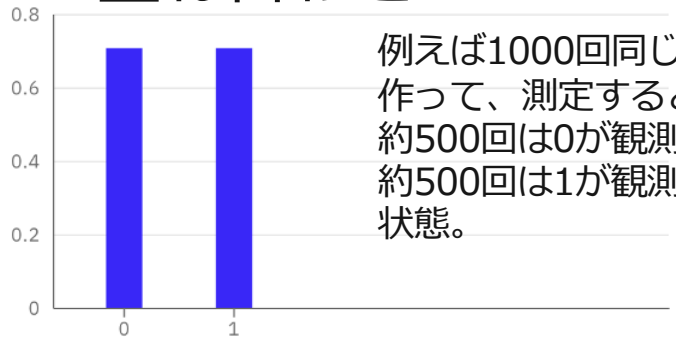
図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

2-1)





重ね合わせ



例えば1000回同じ状態を作って、測定すると約500回は0が観測され、約500回は1が観測される状態。

$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

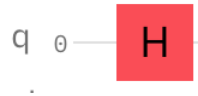
$$0.707 \times |0\rangle + 0.707 \times |1\rangle$$

$$= \frac{1}{\sqrt{2}} \times |0\rangle + \frac{1}{\sqrt{2}} \times |1\rangle$$

2. Hゲート

図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

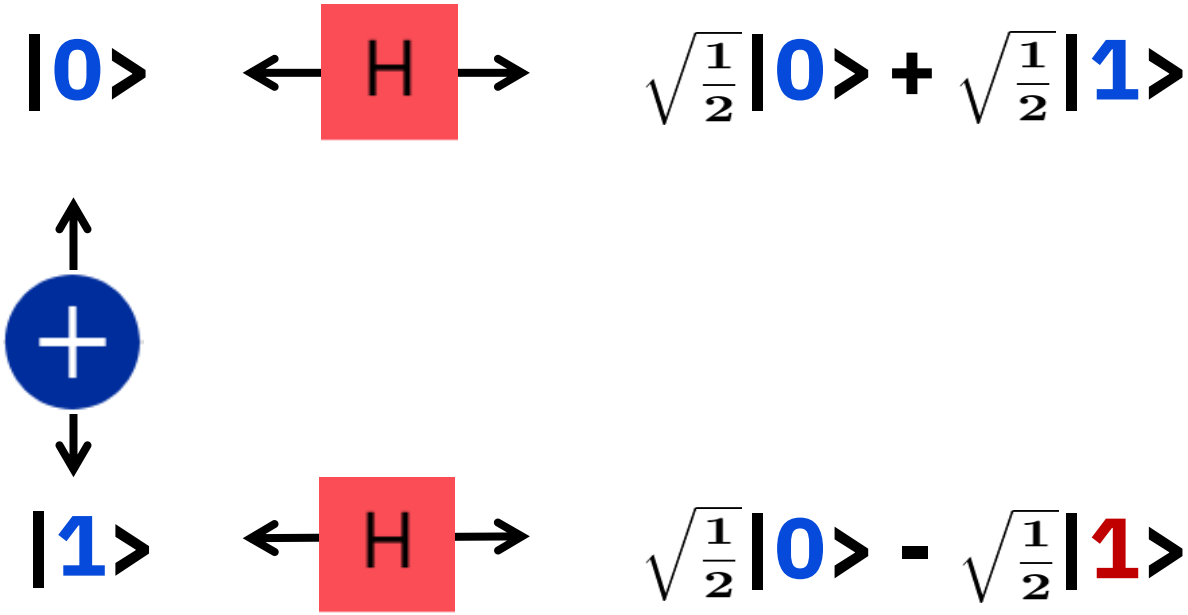
2-1)



2-2)

2-3)

量子コンピューターの計算方法



3. Zゲート

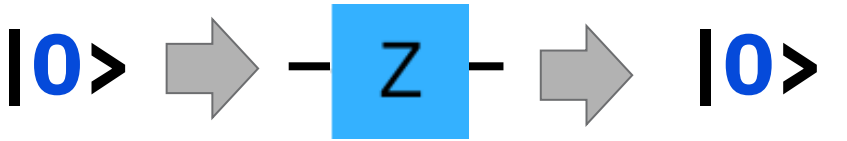
図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

3-1)

3-2)

3-3)

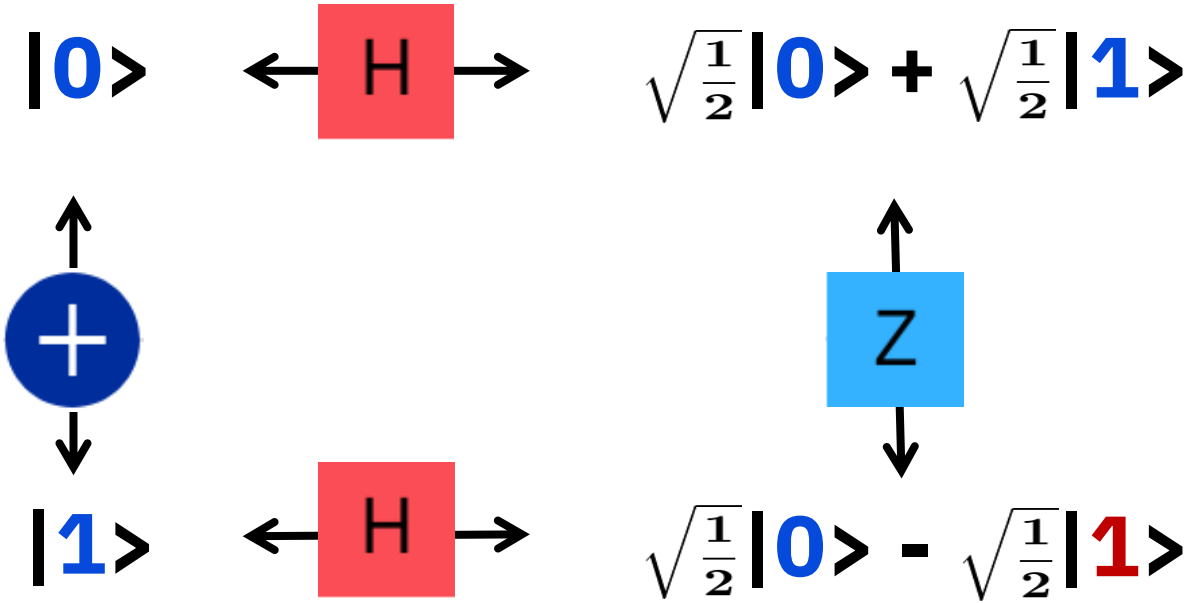
量子コンピューターの計算方法



1 の符号を反転する

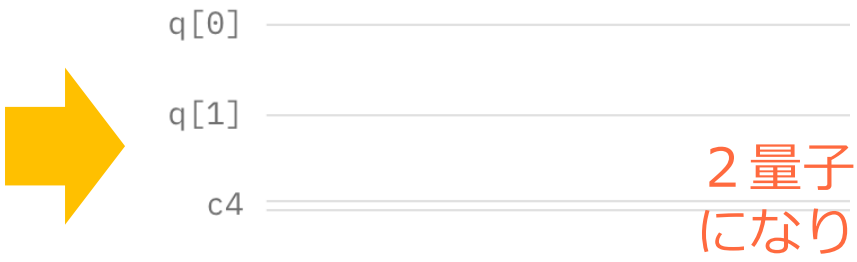
量子コンピューターの計算方法

Qiskit



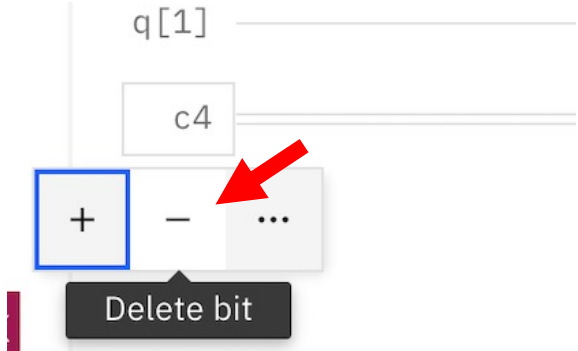
量子ビットを増やす

q[0]をクリックして、さらに「+」マークをクリックして、２量子ビットの回路を準備します。



２量子ビット回路になります

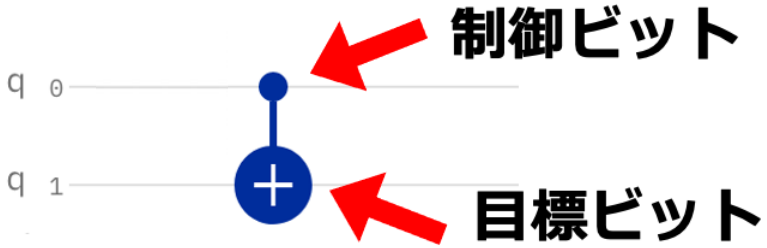
次に、c4をクリックして、「-」マークをクリックするを2回繰り返して、２古典ビットにします。



２古典ビットになります

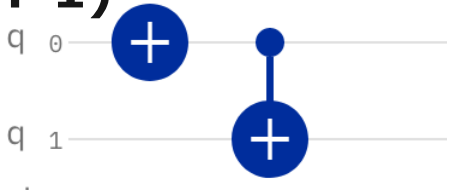
4. CNOTゲート(制御Xゲート)

制御ビットが|1⟩のときのみ、目標ビットを反転（NOT）するゲートです。

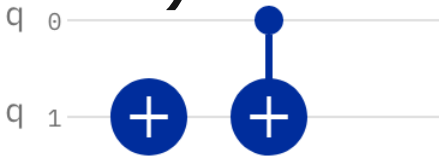


入力		出力	
制御ビット	目標ビット	制御ビット	目標ビット
0	0	0	0
0	1	0	1
1	0	**1**	**1**
1	1	**1**	**0**

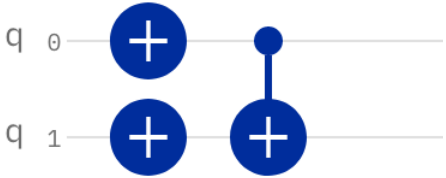
4-1)



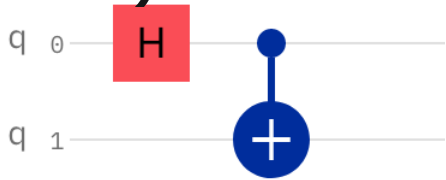
4-2)



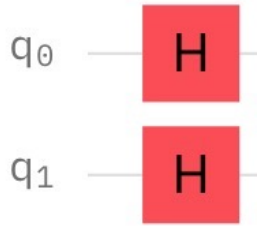
4-3)

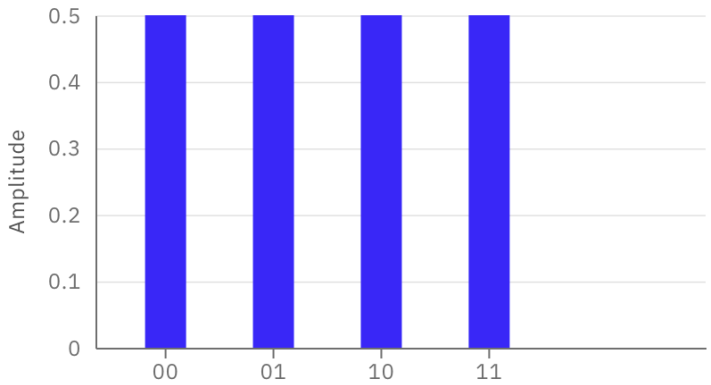


4-4)

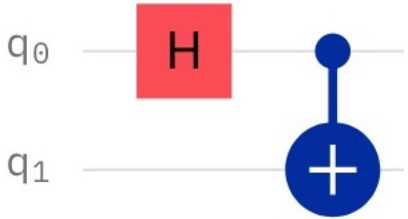


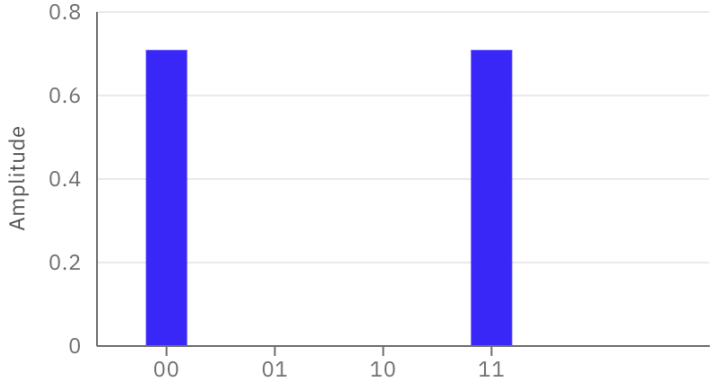
量子重ね合わせ





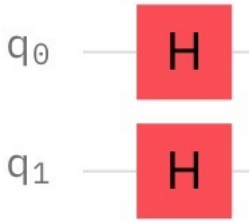
量子もつれ（エンタングルメント）





CNOTゲートは、
エンタングルメントを作ります。

量子重ね合わせ



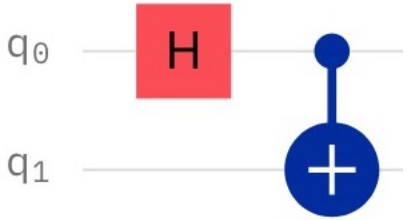
0 0 ••• 25%

0 1 ••• 25%

1 0 ••• 25%

1 1 ••• 25%

量子もつれ（エンタングルメント）



0 0 ••• 50%

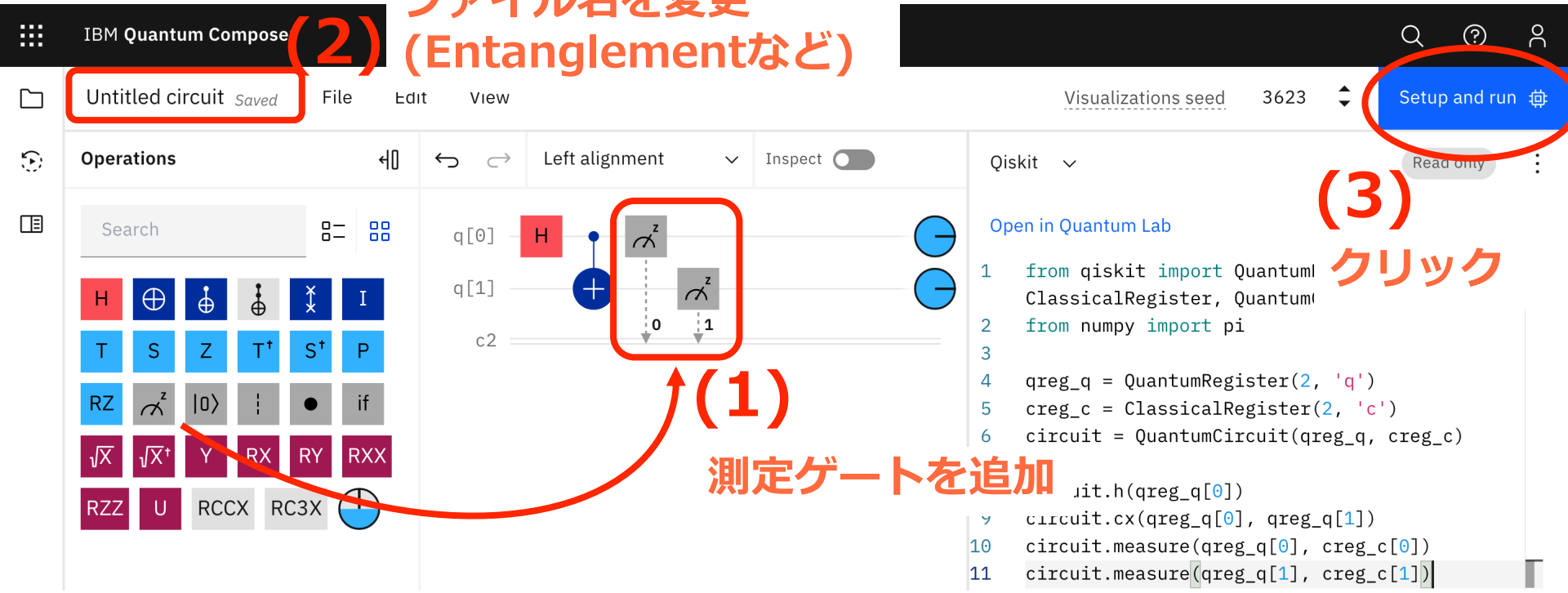
0 1 ••• 0%

1 0 ••• 0%

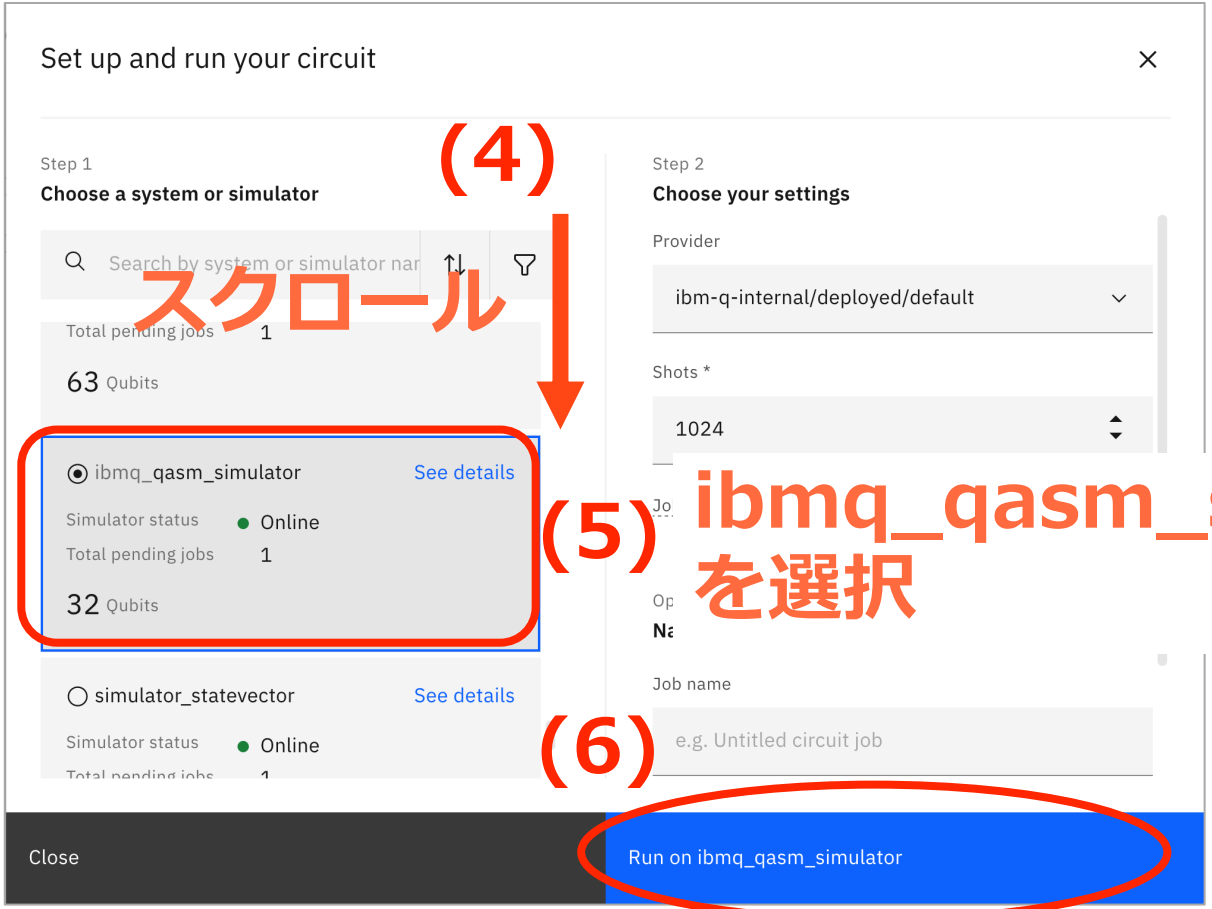
1 1 ••• 50%

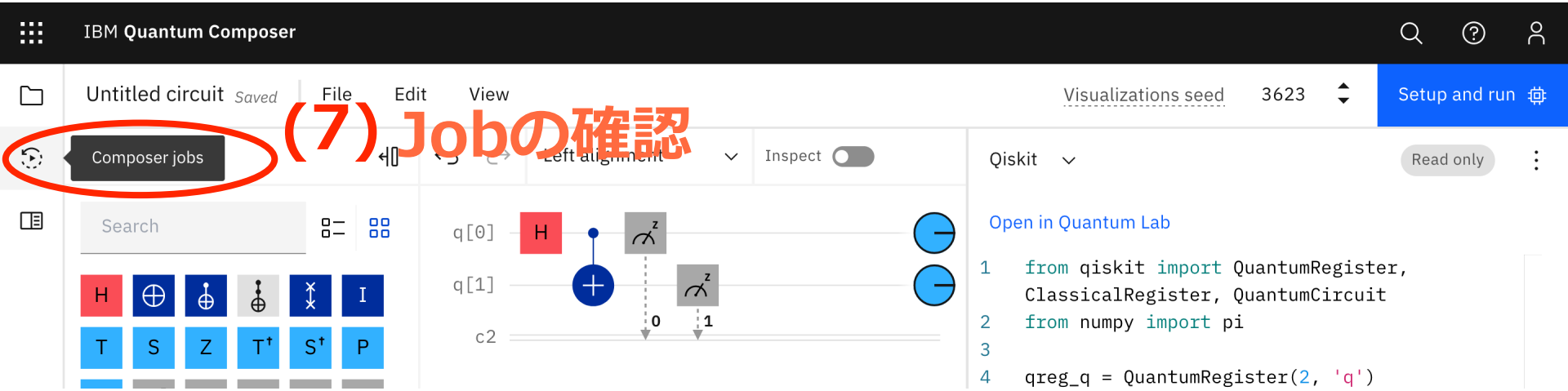
1量子ビット目が0だと分かったら
2量子ビット目も0

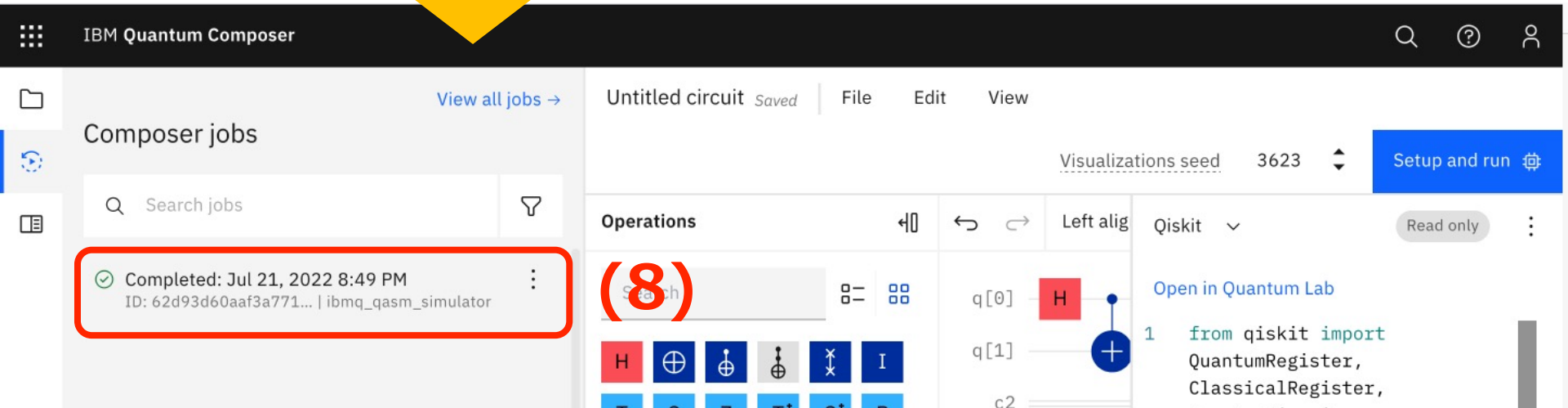
5. 測定


IBM Quantum Compose
(2)
ファイル名を変更
(Entanglementなど)
Untitled circuit Saved
File
Edit
View
Visualizations seed
3623
Setup and run
Operations
Search
H
I
T
S
Z
P
RZ
|0⟩
if
RX
RY
RXX
RZZ
U
RCCX
RC3X
Left alignment
Inspect
q[0]
q[1]
c2
(1)
測定ゲートを追加
Qiskit
Read only
Open in Quantum Lab
(3)
クリック
1 from qiskit import QuantumI
ClassicalRegister, QuantumI
2 from numpy import pi
3
4 qreg_q = QuantumRegister(2, 'q')
5 creg_c = ClassicalRegister(2, 'c')
6 circuit = QuantumCircuit(qreg_q, creg_c)
circuit.h(qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
10 circuit.measure(qreg_q[0], creg_c[0])
11 circuit.measure(qreg_q[1], creg_c[1])

まず量子シミュレーターで実験




IBM Quantum Composer
Untitled circuit Saved
File
Edit
View
Visualizations seed
3623
Setup and run
(7) Jobの確認
Composer jobs
Inspect
Qiskit
Read only
Search
Open in Quantum Lab
q[0]
q[1]
c2
1 from qiskit import QuantumRegister,
ClassicalRegister, QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(2, 'q')


IBM Quantum Composer
View all jobs →
Composer jobs
Search jobs
Completed: Jul 21, 2022 8:49 PM
ID: 62d93d60aaf3a771... | ibmq_qasm_simulator
(8)
Untitled circuit Saved
File
Edit
View
Visualizations seed
3623
Setup and run
Operations
Qiskit
Read only
Open in Quantum Lab
q[0]
q[1]
c2
1 from qiskit import
QuantumRegister,
ClassicalRegister,

IBM Quantum Composer

Jobs /
62d93d60aaf3a771842b9ede

See more details

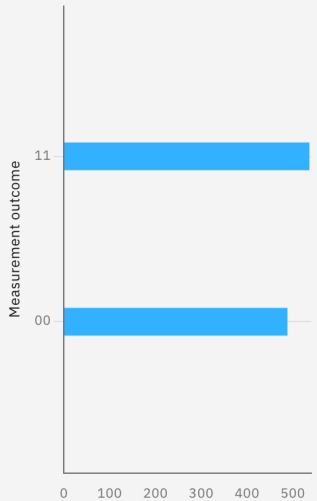
Completed
Jul 21, 2022 8:50 PM (in 8.3s)

Backend
ibmq_qasm_simulator

Status timeline Completed

Details

Result - histogram



(9) 測定結果

Untitled circuit *Saved* | File | Edit | View

Visualizations seed 3623

Setup and run

(10)

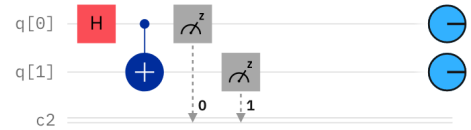
再度実験します

Operations

Search

H I T S Z T† S† P RZ |0⟩ if √X √X† Y RX RY RXX RZZ U RCCX RC3X

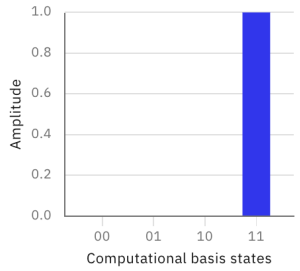
Left alignment | Inspect



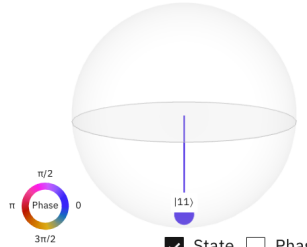
Qiskit | Read only

```
1  from qiskit import QuantumRegister,
   ClassicalRegister, QuantumCircuit
2  from numpy import pi
3
4  qreg_q = QuantumRegister(2, 'q')
5  creg_c = ClassicalRegister(2, 'c')
6  circuit = QuantumCircuit(qreg_q, creg_c)
7
8  circuit.h(qreg_q[0])
9  circuit.cx(qreg_q[0], qreg_q[1])
10 circuit.measure(qreg_q[0], creg_c[0])
11 circuit.measure(qreg_q[1], creg_c[1])
```

Statevector



Q-sphere

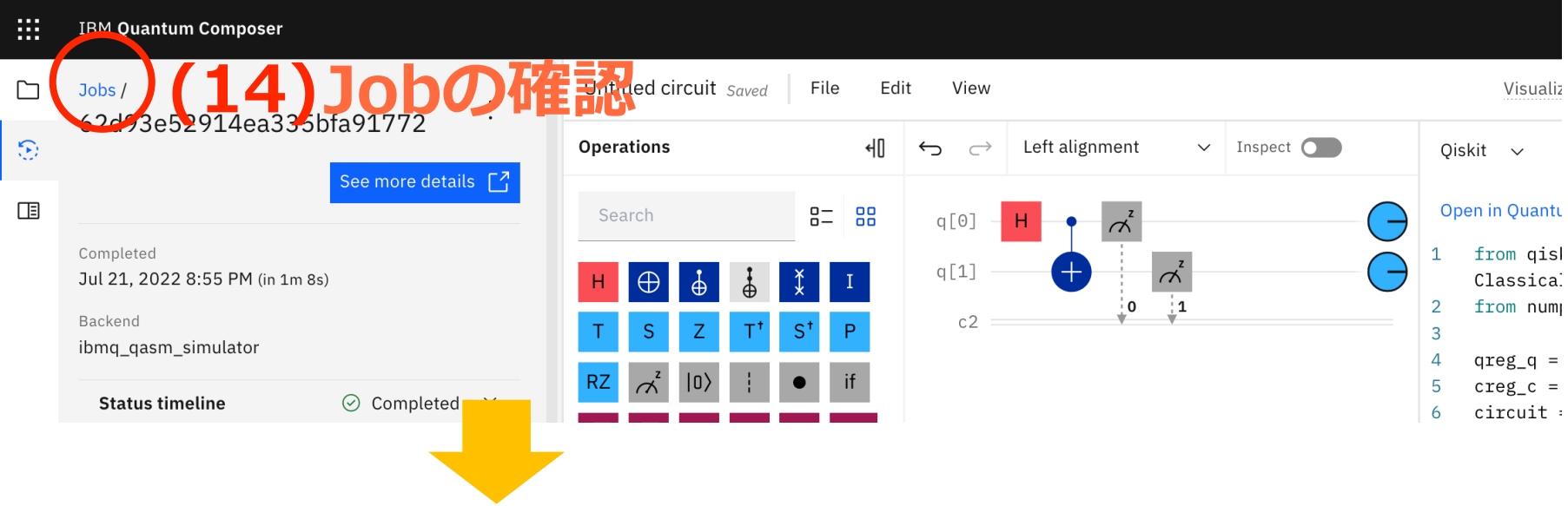


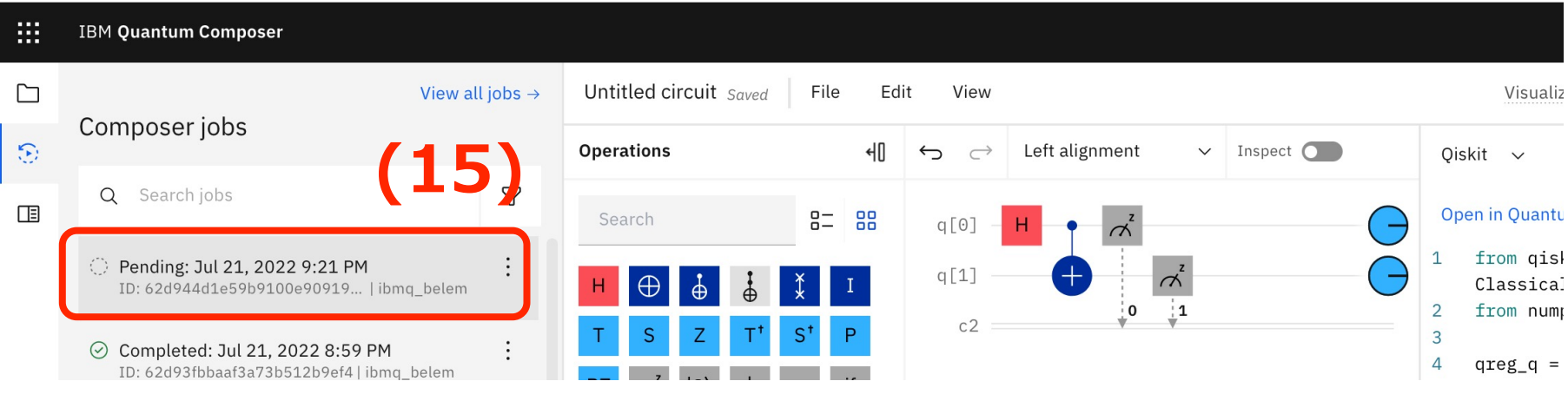
State | Phase angle

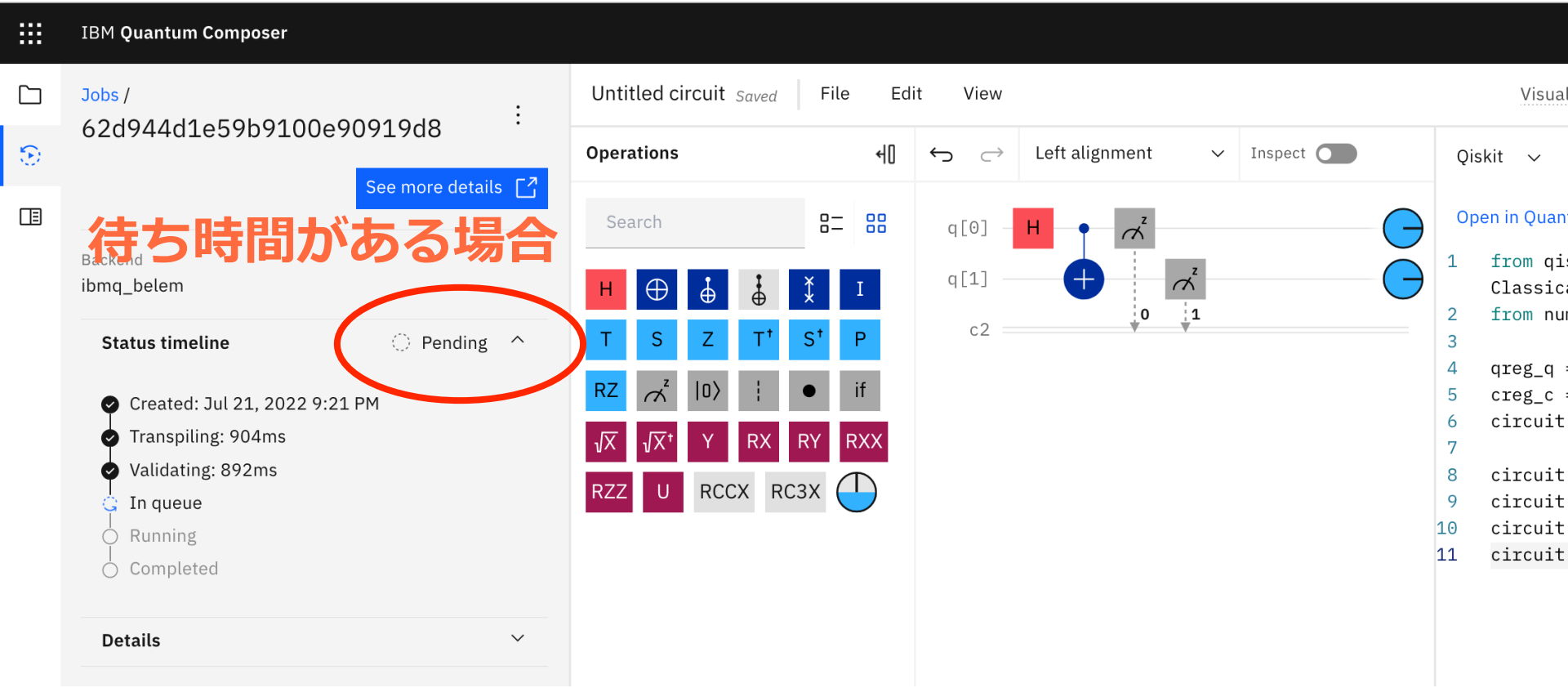
実機の量子コンピューターで実験

Kawasaki-campを選択します

実デバイスを選択します。
・Total pending jobsが少ない
・QVが大きい
などから選んでみましょう。


IBM Quantum Composer
Jobs /
62d93e52914ea335bfa91772
(14) Jobの確認
See more details
Completed
Jul 21, 2022 8:55 PM (in 1m 8s)
Backend
ibmq_qasm_simulator
Status timeline
Completed
Untitled circuit Saved
File
Edit
View
Operations
Search
Left alignment
Inspect
Qiskit
Open in Quantu
q[0]
q[1]
c2
H
I
T
S
Z
P
RZ
if


IBM Quantum Composer
View all jobs →
Composer jobs
(15)
Search jobs
Pending: Jul 21, 2022 9:21 PM
ID: 62d944d1e59b9100e90919... | ibmq_belem
Completed: Jul 21, 2022 8:59 PM
ID: 62d93fbbaaf3a73b512b9ef4 | ibmq_belem
Untitled circuit Saved
File
Edit
View
Operations
Search
Left alignment
Inspect
Qiskit
Open in Quantu
q[0]
q[1]
c2
H
I
T
S
Z
P


IBM Quantum Composer
Jobs /
62d944d1e59b9100e90919d8
See more details
待ち時間がある場合
Backend
ibmq_belem
Status timeline
Pending
Created: Jul 21, 2022 9:21 PM
Transpiling: 904ms
Validating: 892ms
In queue
Running
Completed
Details
Untitled circuit Saved
File
Edit
View
Visual
Operations
Search
H
I
T
S
Z
P
RZ
if
Y
RX
RY
RXX
RZZ
U
RCCX
RC3X
Left alignment
Inspect
q[0]
q[1]
c2
0
1
Qiskit
Open in Quan
1 from qi
Classic
2 from nu
3
4 qreg_q
5 creg_c
6 circuit
7
8 circuit
9 circuit
10 circuit
11 circuit

See more details

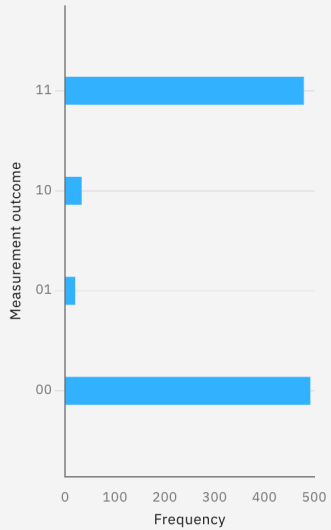
Completed
Jul 21, 2022 9:00 PM (in 38.2s)

Backend
ibmq_belem

Status timeline — Completed

Details

Result - histogram



(16) 実機での結果

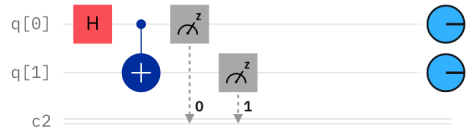
Untitled circuit *Saved* | File | Edit | View | Visualizations seed 3623 | Setup and run

Operations

Search

H ⊕ ● ● ✕ I
T S Z T† S† P
RZ ↺z |0⟩ ┆ ● if
√X √X† Y RX RY RXX
RZZ U RCCX RC3X

Left alignment | Inspect

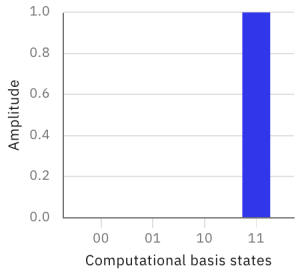


Qiskit | Read only

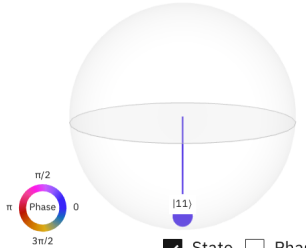
Open in Quantum Lab

```
1  from qiskit import QuantumRegister,
   ClassicalRegister, QuantumCircuit
2  from numpy import pi
3
4  qreg_q = QuantumRegister(2, 'q')
5  creg_c = ClassicalRegister(2, 'c')
6  circuit = QuantumCircuit(qreg_q, creg_c)
7
8  circuit.h(qreg_q[0])
9  circuit.cx(qreg_q[0], qreg_q[1])
10 circuit.measure(qreg_q[0], creg_c[0])
11 circuit.measure(qreg_q[1], creg_c[1])
```

Statevector



Q-sphere

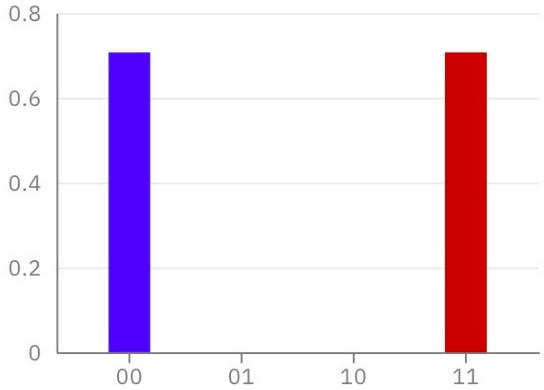


State | Phase angle

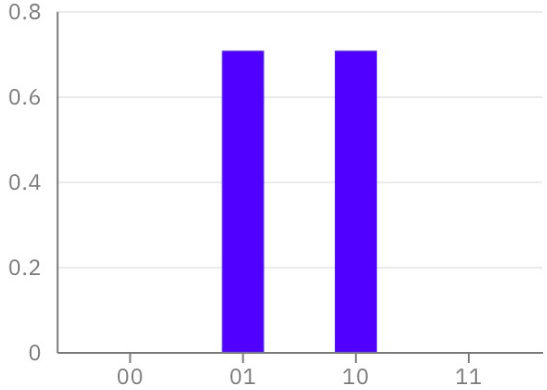
演習問題

２量子ビットのエンタングル状態を作ってみましょう。
答えは一つではないので、どんな作り方でもOKです。

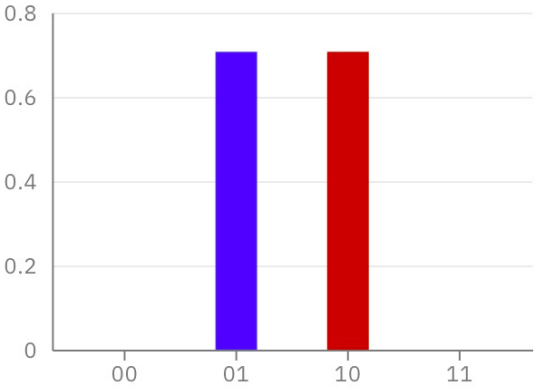
（１）$\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$



（２）$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$



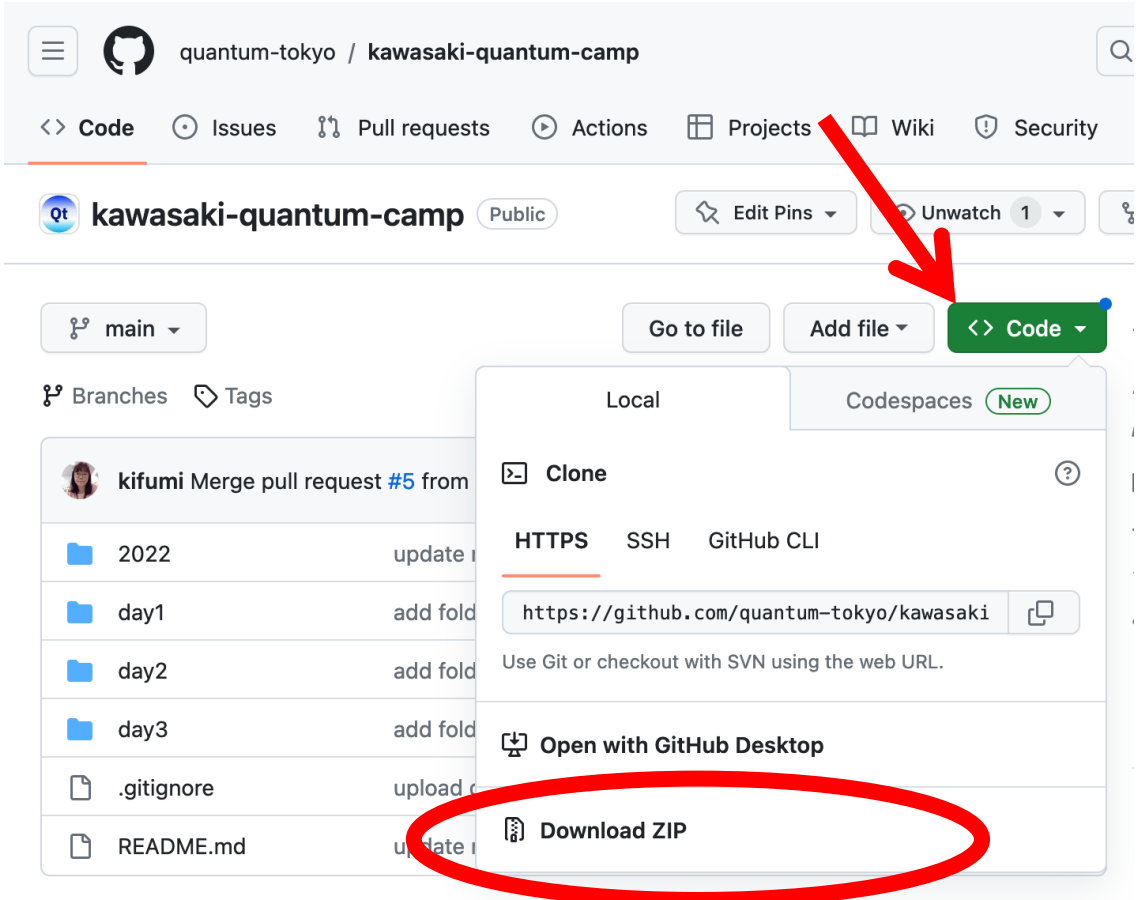
（３）$\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$



ハンズオン ファイルのダウンロード

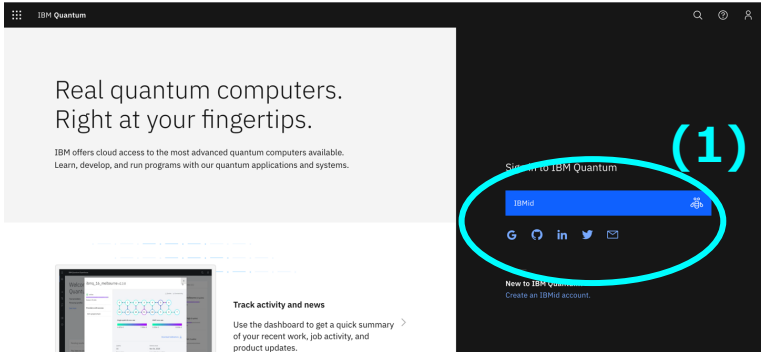
URL: ibm.biz/kwskgit

1. ハンズオンで使用するファイル(zipファイル)をダウンロードします。
2. ダウンロードしたzipファイルを解凍します。
3. 「day1」フォルダーの下のファイルを使います。

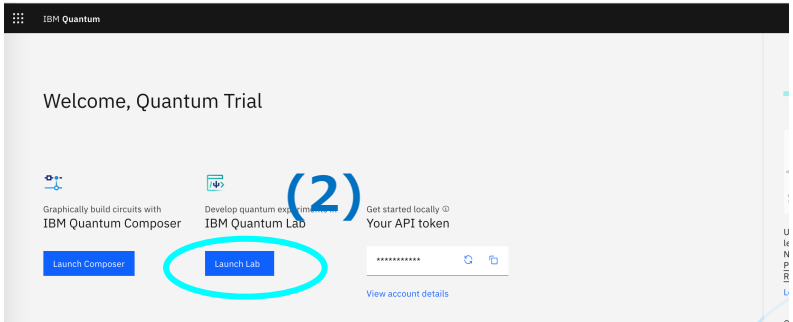


IBM Quantum Labでの実行

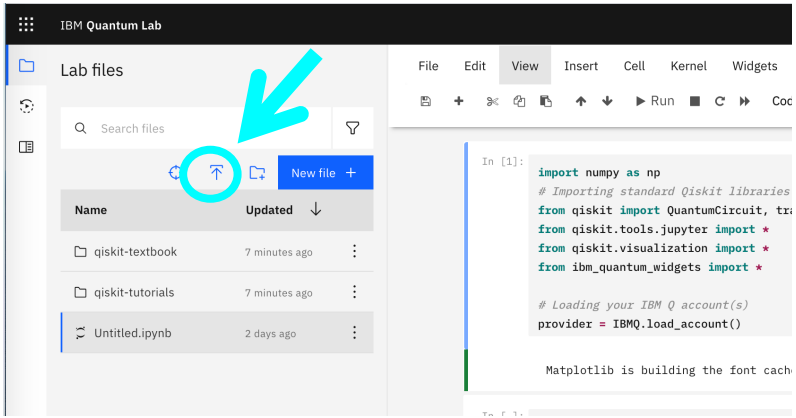
(1) IBM Quantumにログインします。
URL: https://quantum-computing.ibm.com/



(2) 青の右側「Launch Lab」をクリック。



(3)左側  の「Upload file」から、ご自分のローカルに保存したハンズオンコンテンツ「20230817_qiskit.ipynb」を探して、アップロードします。



(4)ファイル名「 20230817_qiskit.ipynb 」をダブルクリックして開きます。