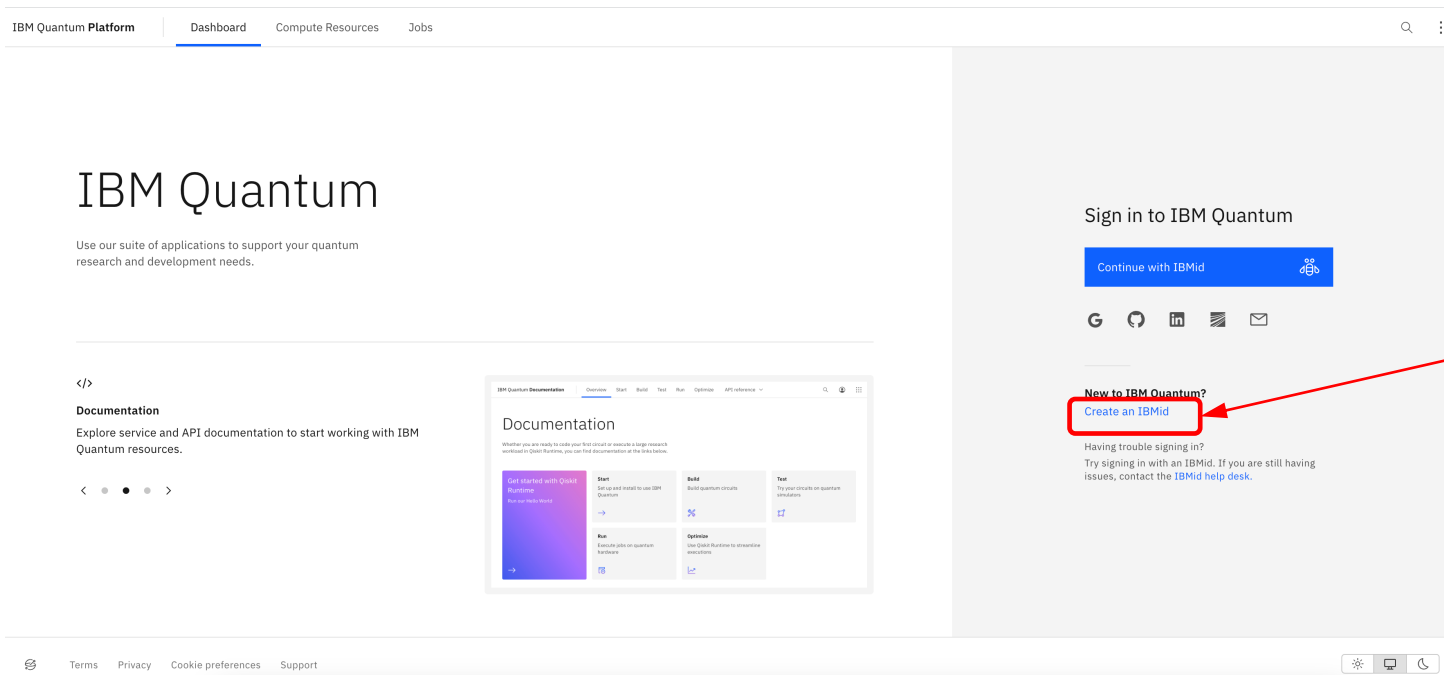


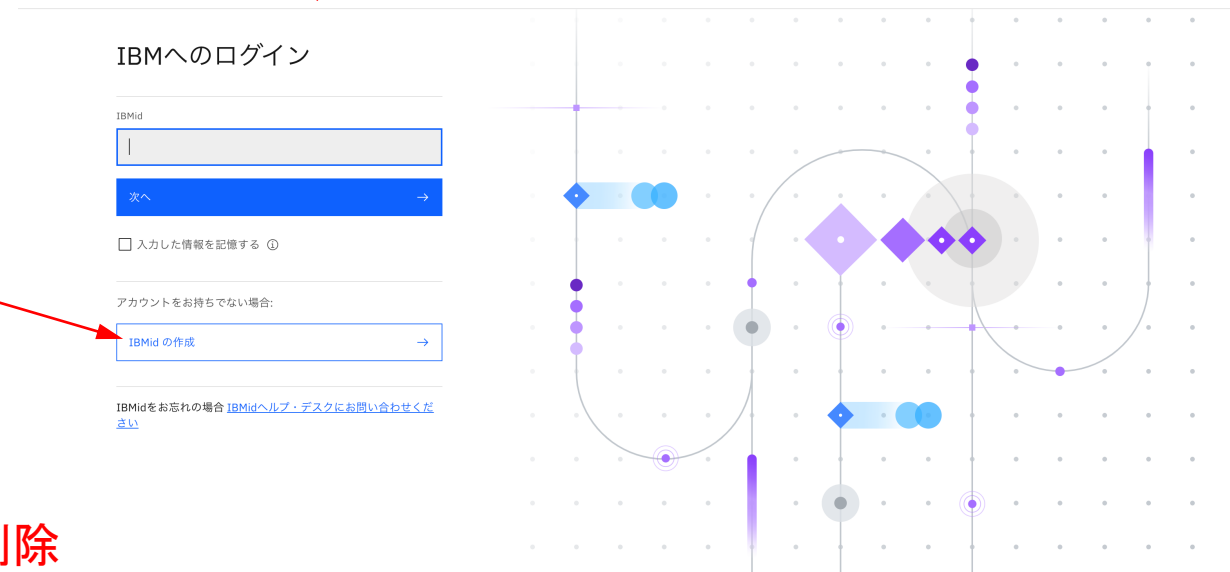
① <https://quantum-computing.ibm.com/> にアクセス



② Create IBMidをクリック



③ IBMidの作成 をクリック
→メールアドレス、氏名などを
入力してアカウント作成

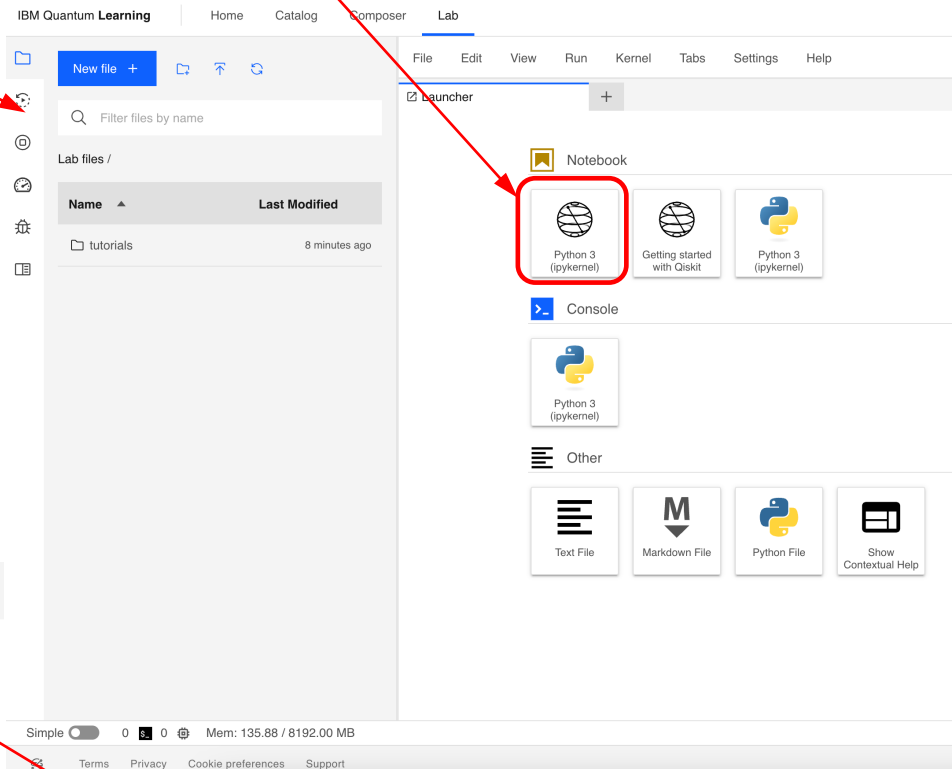
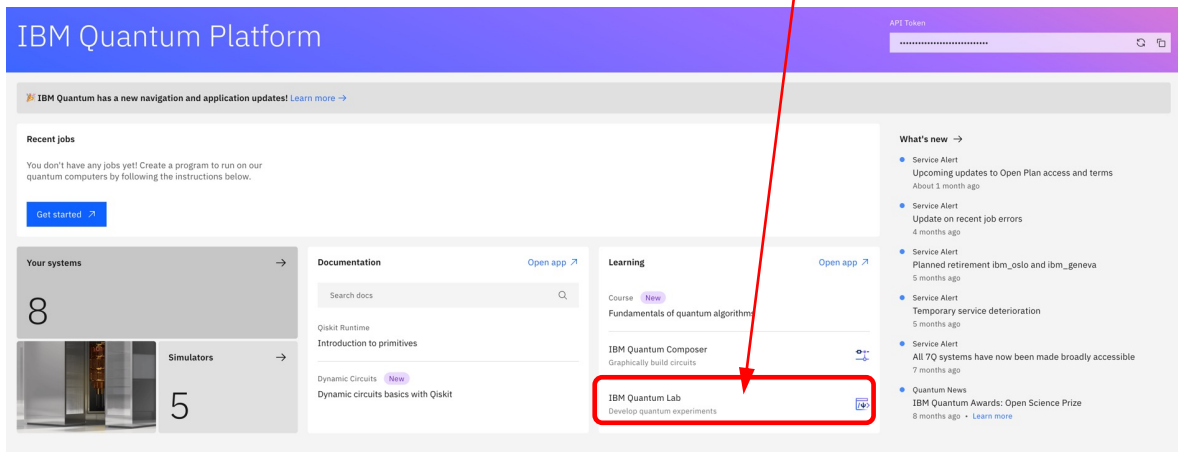


シミュレータだけ使用の場合は無料、
実機を使用する場合は課金が必要

アカウント作成の後に課金するか否かの
ポップアップが表示されるが「×」で表示削除

④ <https://lab.quantum-computing.ibm.com/jupyter>にアクセス
あるいはIBM Quantum PlatformのIBM Quantum Labをクリック

⑤ Python3をクリック



⑥ Jupyterが起動し、以下のコードが出
なければ、以下のコードを入力して
実行Run▶をクリック

```
import qiskit.tools.jupyter
%qiskit_version_table
```

Version Information

Software	Version
qiskit	0.44.0
qiskit-terra	0.25.0
qiskit_aer	0.12.2
qiskit_ibm_runtime	0.11.3
qiskit_ibm_provider	0.6.3

System information

Python version	3.10.8
Python compiler	GCC 10.4.0
Python build	main, Nov 22 2022 08:26:04
OS	Linux
CPUs	8
Memory (Gb)	31.142810821533203
Mon Sep 04 00:39:33 2023 UTC	

⑦ 赤枠のコードを入力して実行
→ライブラリが表示

```
import numpy as np
```

```
#Importing standard Qsskit libraries
```

```
from qiskit import QuantumCircuit, transpile, Aer, IBMQ
```

```
from qiskit.tools.jupyter import *
```

```
from qiskit.visualization import *
```

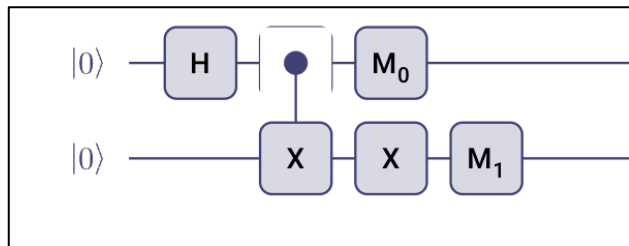
```
from ibm_quantum_widgets import *
```

```
from qiskit.providers.aer import QasmSimulator
```

```
#Loading you IBM Quantum account(s)
```

```
provider = IBMQ.load_account()
```

⑧以下の量子回路を作成して、実行



```
#量子回路の初期化
circuit=QuantumCircuit(2)

#量子回路の組み立て
circuit.h(0) #アダマール行列を使用
circuit.cx(0,1) #CNOTを使用
# 測定
circuit.measure_all()

from qiskit import execute

#実行と結果取得
backend=Aer.get_backend("qasm_simulator") #バックエンドを測定
job=execute(circuit, backend) #量子プログラムを実行
result=job.result() #結果を取得
print(result.get_counts(circuit)) #結果をテキスト表示
```

{'00': 481, '11': 543} **実行結果**

⑨赤枠コード追加して実行 → 出現確率を棒グラフ表示

```
#量子回路の初期化
circuit=QuantumCircuit(2)

#量子回路の組み立て
circuit.h(0) #アダマール行列を使用
circuit.cx(0,1) #CNOTを使用
# 測定
circuit.measure_all()

from qiskit import execute

#実行と結果取得
backend=Aer.get_backend("qasm_simulator") #バックエンドを測定
job=execute(circuit, backend) #量子プログラムを実行
result=job.result() #結果を取得
print(result.get_counts(circuit)) #結果をテキスト表示

plot_distribution(job.result().get_counts(circuit))
```

{'00': 529, '11': 495}

