QISKit cheat sheet

IBM

qubits

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1$$

- Pour chaque état possible la mesure ne peut donner que $|0\rangle$ ou $|1\rangle$
- La probabilité d'obtenir $|0\rangle$ est $|\alpha|^2$, celle d'obtenir $|1\rangle$ est $|\beta|^2$
- Lorsque la mesure est faite, la superposition d'état est perdue et le qubit se retrouve immédiatement dans l'état mesuré.

Le modèle d'un qubit peut être vu comme un espace vectoriel (à coefficients dans $\mathbb{C}$) engendré par les vecteurs de base $|0\rangle$ et $|1\rangle$ (états observables). Les états quantiques sont les vecteurs de norme 1 dans cet espace.

avec 2 qubits : $|\phi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$

avec 3 qubits : $|\phi\rangle = a|000\rangle + b|001\rangle + c|010\rangle + d|011\rangle + e|100\rangle + f|101\rangle + g|110\rangle + h|111\rangle$

... et ainsi de suite : les systèmes de n qubits sont décrits dans un espace vectoriel de dimension 2^n (càd 2^{127} pour 127 qubits)

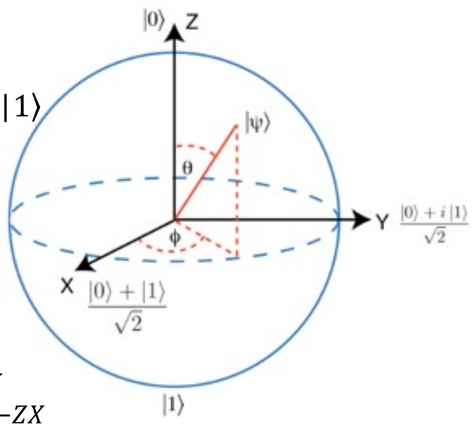
operateurs (gates, portes quantiques)

« PAULI » Operators

rotation around x axis	⊕	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	qc.x(qr[n])	RX	$\begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$
rotation around y axis	Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	qc.y(qr[n])	RY	$\begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$
rotation around z axis	Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	qc.z(qr[n])	RZ	$\begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$
Identity	I	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	qc.id(qr[n])		

La Sphère de Bloch

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$



$$X^2 = Y^2 = Z^2 = I$$
$$XY = iZ\,;\, ZX = iY\,;\, YZ = iX$$
$$XY = -YX\,;\, YZ = -ZY;\, XZ = -ZX$$

superposition $\frac{1}{\sqrt{2}}$(X+Z) Hadamard gate | H | $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ | qc.h(qr[n])

ensemble des portes natives sur les machines d'IBM : CX, ID, RZ, SX, X

CNOT : flips target qubit according to control qubit state.

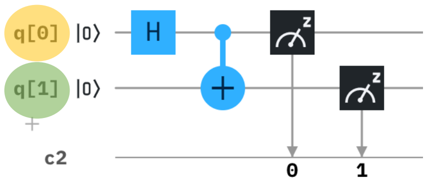
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

measurement measures quantum state in quantum register into classical register (0/1)

de nombreux autres opérateurs sont définis, pour un ou plusieurs qubits : S, T, swap, cswap(Fredkin), ccnot(Toffoli), cz, ...)

circuits

les **circuits** utilisent des **quantum bits ,** q[0] et q[1] initialisés à l'état $|0\rangle$, les **gates** (H, CNOT) sont appliquées en séquence sur les qubits de gauche à droite. A la fin une **mesure** est faite qui produit le résultat dans un **registre classique** (c2 possède 2 bits ici)

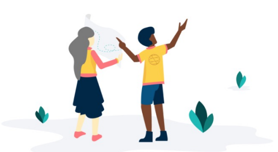


« Bell State »

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

IBM Quantum : https://www.ibm.com/quantum-computing/

IBM Quantum propose des documentations des exemples et des tutoriels. On peut construire des circuits au moyen d'un composeur graphique et les exécuter sur un émulateur local ou distant, ainsi que sur de vrais ordinateurs quantiques gratuitement. On peut aussi utiliser la bibliothèque open source **qiskit** pour Python pour construire et exécuter les programmes sur les ordinateurs d'IBM.



Circuit Composer

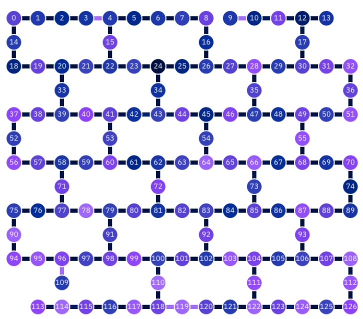
Explore the graphical interface for creating and testing circuits

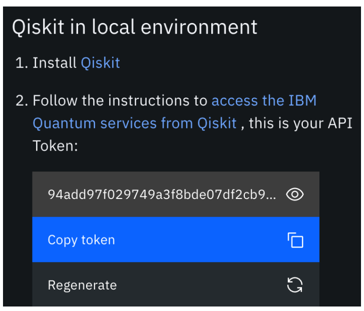
Create a circuit →

Qiskit Notebooks

Create your first notebook and start using Qiskit

Create a notebook →





Visitez

qiskit.org

- Documentation,
- TextBook
- Chaine Youtube
- Slack
- ...

QISKit cheat-sheet

utilisation de qiskit en local, mini rappels conda ou venv

conda : Installer Anaconda (@ anaconda.org) et utiliser l'interface graphique pour créer un environnement conda et y installer qiskit
En mode CLI (qc est le nom de l'environnement choisi par ex.)

- `conda create –n qc python=3`
- `pip install qiskit`
- `pip install –U qiskit` #update
- `conda env remove –n qc`
- `conda env list` (envs) `conda list` (packages)
- `conda activate qc`

venv :
python3 –m venv qc (création d'un environement virtuel)
puis : pip install qiskit, matplotlib, jupyter … et pour activer :
win: **qc\Scripts\activate.bat** **mac:** **source qc/bin/activate**
`python3 -m venv remove qc`

`jupyter notebook`

qiskit : « Hello World! » statevector, émulateur, machine

```
import qiskit ; qiskit.__qiskit_version__
```

```
from qiskit import  QuantumCircuit, execute
from qiskit.visualization import  plot_state_qsphere
from qiskit.quantum_info import Statevector
qc = QuantumCircuit(4)
for i in range(4): qc.h(i)
sv = Statevector.from_label('0'*n)
sv = sv.evolve(qc)
plot_state_qsphere(sv.data,
show_state_labels=True, show_state_phases=True)
```

```
from qiskit import QuantumCircuit, Aer, execute
qc = QuantumCircuit(2,2)
qc.h(0) ; qc.cx(0,1)
qc.measure([0,1],[0,1])
qc.draw(output='mpl')
backend = qiskit.Aer.get_backend('qasm_simulator')
job = qiskit.execute(qc,backend)
print(job.result().get_counts())
```

```
from qiskit_ibm_provider import IBMProvider
# Get your token https://quantum-computing.ibm.com/account
token = "abcdef123…4567890"
provider = IBMProvider(token=token)
#backend = provider.get_backend("simulator_statevector")
backend = provider.get_backend("ibmq_belem")
# Build circuit
from qiskit import QuantumCircuit
circuit = QuantumCircuit(2, 2)
circuit.h(0)
circuit.cx(0,1)
circuit.measure([0,1], [0,1])
# Transpile circuit
from qiskit import transpile
transpiled_circuit = transpile(circuit, backend)
# Run the circuit and get result
job = backend.run(transpiled_circuit)
counts = job.result().get_counts()
print(counts)
```

Postulats (simplifiés) de la mécanique quantique

1- A un système quantique est associé un espace vectoriel $\mathcal{E}$ (sur $\mathbb{C}$) dans lequel les états sont définis par les vecteurs de norme 1 : $|\boldsymbol{\psi}\rangle$
2- A toute grandeur physique mesurable est associée un operateur hermitien Â agissant sur E, il est appelé observable et décrit complètement la grandeur physique associée.
3- La mesure d'une grandeur physique observable *A* ne peut avoir comme résultat que l'une des valeurs propres de l'opérateur Â.
4- cette mesure effectuée sur un système dans l'état initial $|\Psi\rangle$ fournira une valeur a_i valeur propre associée au vecteur propre u_i de Â avec la probabilité $P\Psi(a_i)$, valant le module au carré de la composante de $|\Psi\rangle$ sur u_i
5- Quel que soit l'état initial, si la mesure d'une observable donne le résultat a_i alors immédiatement après la mesure, le vecteur d'état du système se trouve projeté sur le vecteur propre u_i associé à a_i.
6- L'évolution dans le temps de l'état $|\Psi\rangle$ est régie par (eq de Shrödinger):
i h-bar d $|\Psi(t)\rangle$/dt $= H(t)|\Psi(t)\rangle$ *, où H(t) est l'observable correspondant à l'énergie du système.*

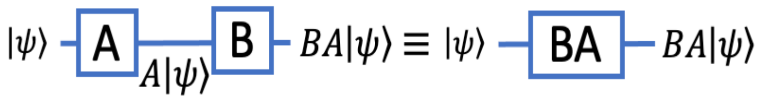
produit hermitien, produit tensoriel, matrices équivalentes à un circuit

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix},\ |\varphi\rangle = \begin{pmatrix}\gamma\\ \delta\end{pmatrix} \quad \langle\psi|\varphi\rangle = (\alpha^* \quad \beta^*)\begin{pmatrix}\gamma\\ \delta\end{pmatrix} = \alpha^*\gamma + \beta^*\delta$$

$$\forall\,\psi :\ \langle\psi|\psi\rangle = \alpha^*\alpha + \beta^*\beta = |\alpha|^2 + |\beta|^2 = 1$$

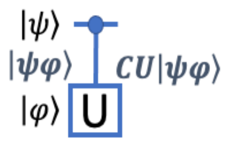
ψ et φ sont orthogonaux $\Leftrightarrow \langle\psi|\varphi\rangle = 0$

$$|\psi_1\rangle\otimes|\psi_2\rangle = \begin{pmatrix}\alpha_1\\ \beta_1\end{pmatrix}\otimes\begin{pmatrix}\alpha_2\\ \beta_2\end{pmatrix} = \begin{pmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{pmatrix} \quad |\psi_1\rangle\otimes|\psi_2\rangle\otimes|\psi_3\rangle = \begin{pmatrix}\alpha_1\\ \beta_1\end{pmatrix}\otimes\begin{pmatrix}\alpha_2\\ \beta_2\end{pmatrix}\otimes\begin{pmatrix}\alpha_3\\ \beta_3\end{pmatrix} = \begin{pmatrix}\alpha_1\alpha_2\alpha_3\\ \alpha_1\alpha_2\beta_3\\ \alpha_1\beta_2\alpha_3\\ \alpha_1\beta_2\beta_3\\ \beta_1\alpha_2\alpha_3\\ \beta_1\alpha_2\beta_3\\ \beta_1\beta_2\alpha_3\\ \beta_1\beta_2\beta_3\end{pmatrix}$$



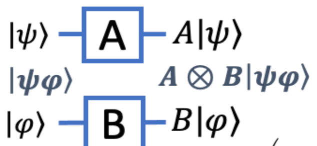
$$BA = \begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix}\times\begin{pmatrix}a_{11} & a_{21}\\ a_{21} & a_{22}\end{pmatrix} = \begin{pmatrix}a_{11}b_{11} + a_{21}b_{12} & a_{21}b_{11} + a_{22}b_{12}\\ a_{11}b_{21} + a_{21}b_{22} & a_{21}b_{21} + a_{21}b_{22}\end{pmatrix}$$

$$CX = \begin{pmatrix}I_2 & 0\\ 0 & X\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

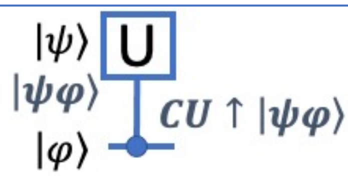


$$CU = \begin{pmatrix}I_2 & 0\\ 0 & U\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & u_{11} & u_{12}\\ 0 & 0 & u_{21} & u_{22}\end{pmatrix}$$

$$CZ = \begin{pmatrix}I_2 & 0\\ 0 & Z\end{pmatrix} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{pmatrix}$$



$$A\otimes B = \begin{pmatrix}a_{11}B & a_{12}B\\ a_{21}B & a_{22}B\end{pmatrix} = \begin{pmatrix}a_{11}\begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix} & a_{12}\begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix}\\ a_{21}\begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix} & a_{22}\begin{pmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\end{pmatrix}\end{pmatrix} = \begin{pmatrix}a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12}\\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22}\\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12}\\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22}\end{pmatrix}$$



$$CU\uparrow = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & u_{11} & 0 & u_{21}\\ 0 & 0 & 1 & 0\\ 0 & u_{12} & 0 & u_{22}\end{pmatrix}$$