What is Qiskit?

- The name "**Qiskit**" is a general term referring to a collection of software for executing programs on quantum computers.

- Most notably among these software tools is the open-source **Qiskit SDK**, and the runtime environment (accessed using **Qiskit Runtime**) through which you can execute workloads on IBM's quantum computers.

- There are also many open-source projects that are part of the broader Qiskit ecosystem (such as Qiskit Optimization, Qiskit Machine Learning, Qiskit Finance etc.). It is important to note that these are maintained by the community, not IBM directly.
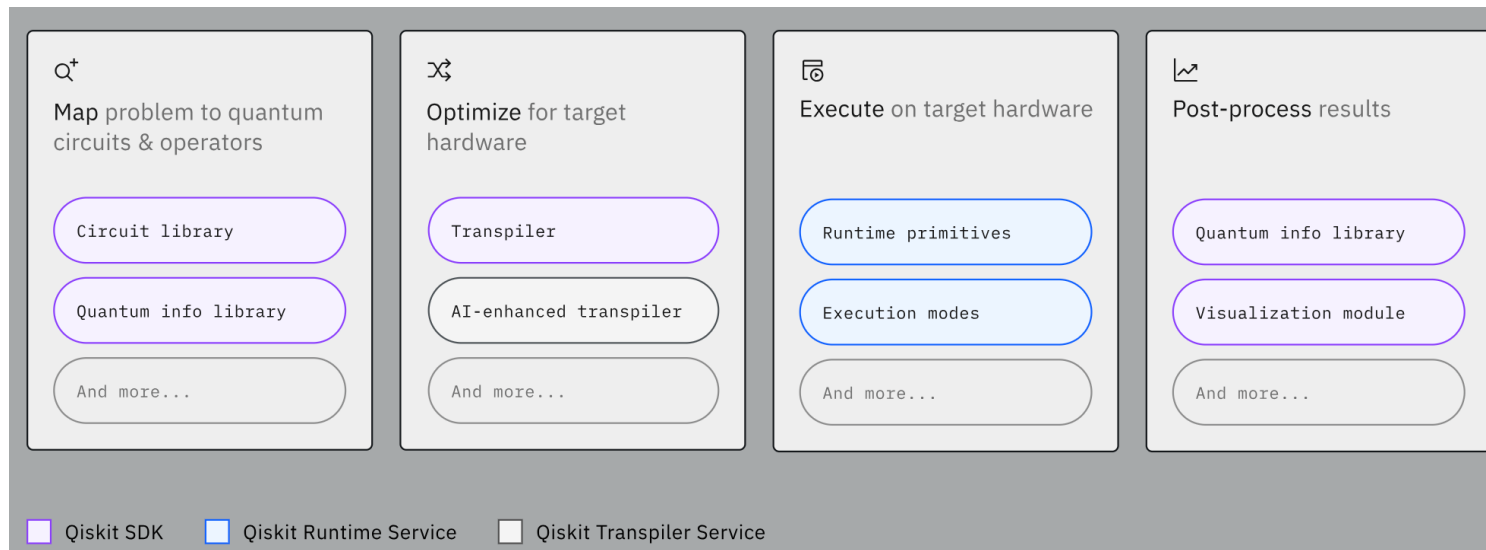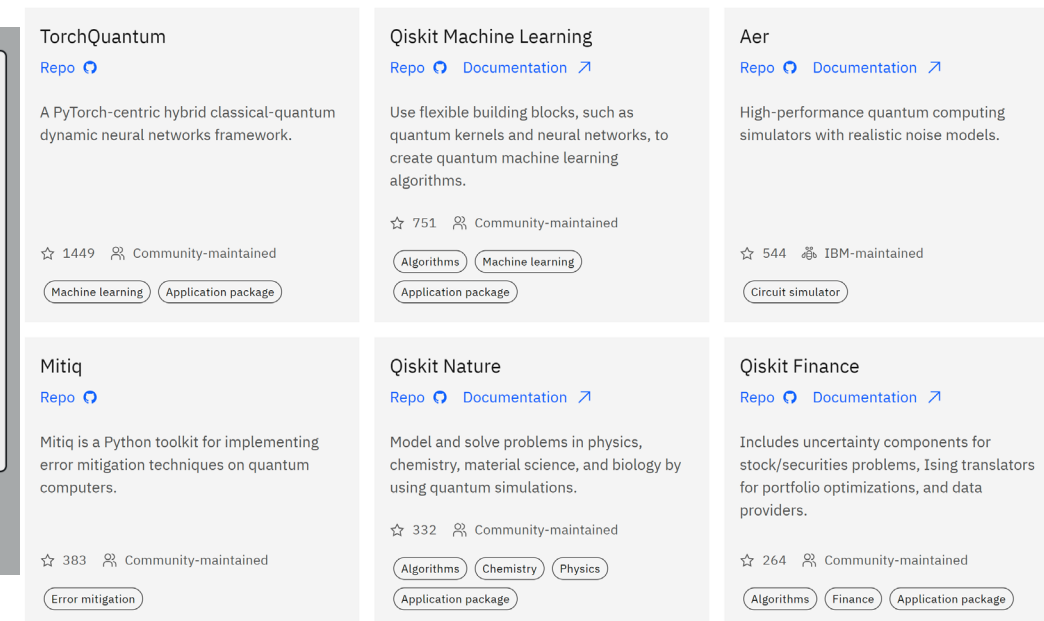

Map problem to quantum circuits & operators
- Circuit library
- Quantum info library
- And more...

Optimize for target hardware
- Transpiler
- AI-enhanced transpiler
- And more...

Execute on target hardware
- Runtime primitives
- Execution modes
- And more...

Post-process results
- Quantum info library
- Visualization module
- And more...

□ Qiskit SDK   □ Qiskit Runtime Service   □ Qiskit Transpiler Service

Image from: https://docs.quantum.ibm.com/guides

TorchQuantum
Repo
A PyTorch-centric hybrid classical-quantum dynamic neural networks framework.
☆ 1449   Community-maintained
Machine learning   Application package

Qiskit Machine Learning
Repo   Documentation
Use flexible building blocks, such as quantum kernels and neural networks, to create quantum machine learning algorithms.
☆ 751   Community-maintained
Algorithms   Machine learning
Application package

Aer
Repo   Documentation
High-performance quantum computing simulators with realistic noise models.
☆ 544   IBM-maintained
Circuit simulator

Mitiq
Repo
Mitiq is a Python toolkit for implementing error mitigation techniques on quantum computers.
☆ 383   Community-maintained
Error mitigation

Qiskit Nature
Repo   Documentation
Model and solve problems in physics, chemistry, material science, and biology by using quantum simulations.
☆ 332   Community-maintained
Algorithms   Chemistry   Physics
Application package

Qiskit Finance
Repo   Documentation
Includes uncertainty components for stock/securities problems, Ising translators for portfolio optimizations, and data providers.
☆ 264   Community-maintained
Algorithms   Finance   Application package

Image from: https://www.ibm.com/quantum/ecosystem

Performant software for best QC capabilities

- In the year 2024, Qiskit SDK was one of the focus areas for IBM Quantum. With the release of Qiskit 1.x.x (and now Qiskit 2.x.x), a comprehensive benchmarking was performed to compare different SDKs.

Benchmarking the performance of quantum computing software

Paul D. Nation,[1, *] Abdullah Ash Saki,[1] Sebastian Brandhofer,[2] Luciano Bello,[3] Shelly Garion,[4] Matthew Treinish,[1] and Ali Javadi-Abhari[1]

[1] IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598 USA
[2] IBM Quantum, IBM Germany Research & Development, Böblingen Germany
[3] IBM Quantum, IBM Research Europe, Zurich Switzerland
[4] IBM Quantum, IBM Research Israel, Haifa 3498825, Israel
(Dated: February 18, 2025)

We present Benchpress, a benchmarking suite for evaluating the performance and range of functionality of multiple quantum computing software development kits. This suite consists of a collection of over 1000 tests measuring key performance metrics for a wide variety of operations on quantum circuits comprised of up to 930 qubits and $\mathcal{O}(10^6)$ two-qubit gates, as well as an execution framework for running the tests over multiple quantum software packages in a unified manner. We give a detailed overview of the benchmark suite, its methodology, and generate representative results over seven different quantum software packages. The flexibility of the Benchpress framework allows for benchmarking that not only keeps pace with quantum hardware improvements but can preemptively gauge the quantum circuit processing costs of future device architectures. Being open-source, Benchpress ensures the transparency and verification of performance claims.

a)
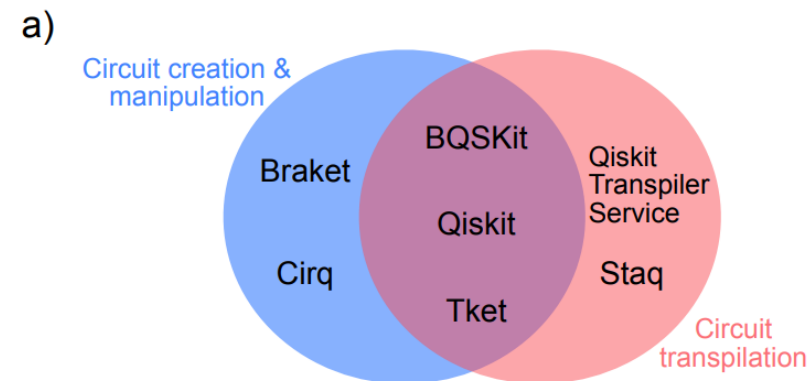
b)

	PASSED	SKIPPED	FAILED	XFAIL
BQSKit	841	22	201	2
Braket	7	1057	2	0
Cirq	10	1054	2	0
Qiskit	1044	22	0	0
QTS	1013	34	19	0
Staq	549	515	2	0
Tket	957	22	87	0

Setting up Qiskit env and IBM Quantum Platform

- First, you need to install Qiskit and Qiskit Runtime. The requirement.txt file in the qml4omics github repo contains the necessary installations. (Follow the steps here https://github.com/IBM/qml4omics/tree/Tutorial25)

- Now, you need to create an account on IBM Quantum Platform (IQP). I recommend using your CCF email address, especially for access to ibm_cleveland quantum computer in the future. (Link: https://quantum.ibm.com/login)

- Your user account is associated with one or more instances (in the form hub / group / project) that give access to IBM Quantum services. Additionally, a unique token is assigned to each account, allowing for IBM Quantum access from Qiskit. After you login, copy that API token.

- Now, in your Python environment you created, you need to run the following (either in a script or a notebook) to save your credentials. If this is not a trusted device, then you need to copy your API key for login. (link to instructions: https://docs.quantum.ibm.com/guides/setup-channel)

```
token = "<your-token>"

1   from qiskit_ibm_runtime import QiskitRuntimeService
2
3   QiskitRuntimeService.save_account(
4       token=token,
5       channel="ibm_quantum" # `channel` distinguishes between different account types
6   )
```