

**Qiskit** is an open-source software development kit (SDK) for working with quantum computers at the level of circuits, pulses, and algorithms. It provides tools for creating and manipulating quantum programs and running them on prototype quantum devices on IBM Quantum Platform or on simulators on a local computer. It follows the circuit model for universal quantum computation, and can be used for any quantum hardware (currently supports superconducting qubits and trapped ions)<sup>[4]</sup> that follows this model.

Qiskit was founded by IBM Research to allow software development for their cloud quantum computing service, IBM Quantum Experience.<sup>[5][6]</sup> Contributions are also made by external supporters, typically from academic institutions.<sup>[7][8]</sup>

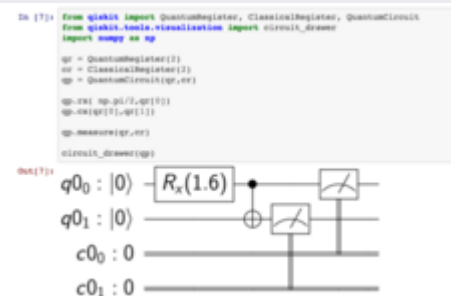
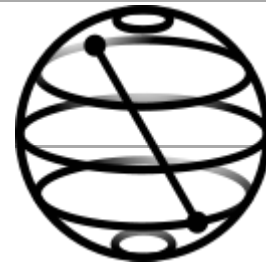
The primary version of Qiskit uses the Python programming language. Versions for Swift<sup>[9]</sup> and JavaScript<sup>[10]</sup> were initially explored, though the development for these versions has halted. Instead, a minimal re-implementation of basic features is available as *MicroQiskit*,<sup>[11]</sup> which is made to be easy to port to alternative platforms.

A range of Jupyter notebooks are provided with examples of quantum computing being used.<sup>[12]</sup> Examples include the source code behind scientific studies that use Qiskit,<sup>[13]</sup> as well as a set of exercises to help people to learn the basics of quantum programming. An open source textbook based on Qiskit is available as a university-level quantum algorithms or quantum computation course supplement.<sup>[14]</sup>

## Components

Qiskit is made of elements that work together to enable quantum computing. The central goal of Qiskit is to build a software stack that makes it easier for anyone to use quantum computers, regardless of their skill level or area of interest; Qiskit allows users to design experiments and applications and run them on real quantum computers and/or classical simulators. Qiskit provides the

## Qiskit



<b>Developer(s)</b>	<u>IBM Research</u> , Qiskit community
<b>Initial release</b>	March 7, 2017. <sup>[1]</sup>
<b>Stable release</b>	0.45.0 / 3 November 2023 <sup>[2]</sup>
<b>Repository</b>	<u>github.com</u> <u>/Qiskit/qiskit</u> ( <u>https://github.com/Qiskit/qiskit</u> )
<b>Written in</b>	<u>Python</u>
<b>Operating system</b>	<u>Cross-platform</u>
<b>Type</b>	<u>SDK</u> for <u>Quantum</u> <u>Computing</u>
<b>License</b>	<u>Apache</u> <u>License 2.0</u> <sup>[3]</sup>
<b>Website</b>	<u>qiskit.org</u> ( <u>https://qiskit.org</u> )

ability to develop quantum software both at the machine code level of OpenQASM, and at abstract levels suitable for end-users without quantum computing expertise. This functionality is provided by the following components.<sup>[15]</sup>

## Qiskit Terra

The element Terra is the foundation on which the rest of Qiskit is built. Qiskit Terra provides tools to create quantum circuits at or close to the level of quantum machine code.<sup>[16]</sup> It allows the processes that run on quantum hardware to be explicitly constructed in terms of quantum gates. It also provides tools to allow quantum circuits to be optimized for a device, as well as managing batches of jobs and running them on remote-access quantum devices and simulators.

The following shows a simple example of Qiskit Terra. In this, a quantum circuit is created for two qubits, which consists of the quantum gates required to create a Bell state. The quantum circuit then ends with quantum measurements, which extract a bit from each qubit.

```
from qiskit import QuantumCircuit

qc = QuantumCircuit(2, 2)

qc.h(0)
qc.cx(0, 1)
qc.measure([0,1], [0,1])
```

## Qiskit Aer

The element Aer provides high-performance quantum computing simulators with realistic noise models. In the near-term, development of quantum software will depend largely on simulation of small quantum devices. For Qiskit, this is provided by the Aer component. This provides simulators hosted locally on the user's device, as well as HPC resources available through the cloud.<sup>[17]</sup> The simulators can also simulate the effects of noise for simple and sophisticated noise models.<sup>[18]</sup>

Continuing with the previous example: Once the quantum circuit has been created, it can be run on a backend (either quantum hardware or a simulator). In the following example, a local simulator is used.

```
from qiskit import Aer, execute

backend = Aer.get_backend("qasm_simulator")
job = execute(qc, backend)
result = job.result()
print(result.get_counts(qc))
```

The final print statement here will show the results returned by the backend. This is a Python dictionary that describes the bit strings obtained from multiple runs of the quantum circuit. In the quantum circuit used in this example, the bit strings '00' and '11' should be the only possible results, and should occur with equal probability. The full results will therefore typically have the samples split approximately equally between the two, such as {'00': 519, '11': 505} .

Experiments done on quantum hardware using Qiskit have been used in many research papers,<sup>[19]</sup> such as in tests of quantum error correction,<sup>[20][21]</sup> generation of entanglement<sup>[22]</sup> and simulation of far-from-equilibrium dynamics.<sup>[23]</sup>

## Qiskit Ignis

*As of version 0.7.0, released on 6th December 2021, Qiskit Ignis has been deprecated and superseded by the Qiskit Experiments project.*<sup>[24]</sup>

The element Ignis provides tools for quantum hardware verification, noise characterization, and error correction. Ignis is a component that contains tools for characterizing noise in near-term devices, as well as allowing computations to be performed in the presence of noise. This includes tools for benchmarking near-term devices, error mitigation and error correction.<sup>[25]</sup>

Ignis is meant for those who want to design quantum error correction codes, or who wish to study ways to characterize errors through methods such as tomography, or even to find a better way for using gates by exploring dynamical decoupling and optimal control.

## Qiskit Aqua

*As of version 0.9.0, released on 2nd April 2021, Qiskit Aqua has been deprecated with its support ending and eventual archival being no sooner than 3 months from that date.*

The element Aqua provided a library of cross-domain algorithms upon which domain-specific applications can be built. However, the Qiskit 0.25.0 release included a restructuring of the applications and algorithms. What previously has been referred to as Qiskit Aqua, the single applications and algorithms module of Qiskit, is now split into dedicated application modules for Optimization, Finance, Machine Learning and Nature (including Physics & Chemistry). The core algorithms and opflow operator functionality were moved to Qiskit Terra.

Additionally, to the restructuring, all algorithms follow a unified paradigm: algorithms are classified according to the problems they solve, and within one application class algorithms can be used interchangeably to solve the same problem. This means that, unlike before, algorithm instances are decoupled from the problem they solve.<sup>[26]</sup>

## Qiskit Optimization

Qiskit Optimization is an open-source framework that covers the whole range from high-level modeling of optimization problems, with automatic conversion of problems to different required representations, to a suite of easy-to-use quantum optimization algorithms that are ready to run on classical simulators, as well as on real quantum devices via Qiskit. The Optimization module enables easy, efficient modeling of optimization problems using docplex (<https://developer.ibm.com/docloud/documentation/optimization-modeling/modeling-for-python/>).<sup>[27]</sup>

## Qiskit Finance

Qiskit Finance is an open-source framework that contains uncertainty components for stock/securities problems, Ising translators for portfolio optimizations and data providers to source real or random data to finance experiments.<sup>[28]</sup>

## Qiskit Machine Learning

The Machine Learning package (as of 2021) contains sample datasets at present. It has some classification algorithms such as QSVM and VQC (Variational Quantum Classifier), where this data can be used for experiments, and there is also QGAN (Quantum Generative Adversarial Network) algorithm.<sup>[29]</sup>

## Qiskit Nature

Qiskit Nature is an open-source framework that supports problems including ground state energy computations, excited states and dipole moments of molecule, both open and closed-shell. The code comprises chemistry drivers, which when provided with a molecular configuration will return one and two-body integrals as well as other data that is efficiently computed classically. This output data from a driver can then be used as input in Qiskit Nature that contains logic which is able to translate this into a form that is suitable for quantum algorithms.<sup>[30]</sup>

## See also

---

- IBM Quantum Platform
- Cirq, Google's Quantum framework
- Quantum programming

## References

---

1. Jay M. Gambetta; Andrew Cross (March 27, 2018). "Looking back on a year of Qiskit" (<https://medium.com/qiskit/qiskit-turns-one-looking-back-cbc2c48d7a95>). *Medium*. Archived (<https://web.archive.org/web/20211216225930/https://medium.com/qiskit/qiskit-turns-one-looking-back-cbc2c48d7a95>) from the original on December 16, 2021. Retrieved September 24, 2019.
2. "Releases – Qiskit" (<https://github.com/Qiskit/qiskit/releases>). *GitHub*. Archived (<https://web.archive.org/web/20211216225948/https://github.com/Qiskit/qiskit/releases>) from the original on 2021-12-16. Retrieved 2019-12-20.
3. License (<https://github.com/Qiskit/qiskit-terra/blob/master/LICENSE.txt>)
4. "Qiskit - Write once, target multiple architectures" (<https://www.ibm.com/blogs/research/2019/11/qiskit-for-multiple-architectures/>). *IBM Research Blog*. 2019-11-05. Archived (<https://web.archive.org/web/20201126152348/https://www.ibm.com/blogs/research/2019/11/qiskit-for-multiple-architectures/>) from the original on 2020-11-26. Retrieved 2019-12-20.
5. Magee, Tamlim (August 24, 2018). "What is Qiskit, IBM's open source quantum computing framework" (<https://web.archive.org/web/20190306044548/https://www.computerworlduk.com/open-source/inside-qiskit-ibms-open-source-quantum-computing-framework-3682891/>). *Computerworld UK*. Archived from the original (<https://www.computerworlduk.com/open-source/inside-qiskit-ibms-open-source-quantum-computing-framework-3682891/>) on 6 March 2019. Retrieved 11 December 2018.

6. Hemsoth, Nicole (August 7, 2018). "QISKit Developments Key to IBM Quantum Engagement" (<https://www.nextplatform.com/2018/08/07/qiskit-developments-key-to-ibm-quantum-engagement/>). *The Next Platform*. Archived (<https://web.archive.org/web/20190306042859/https://www.nextplatform.com/2018/08/07/qiskit-developments-key-to-ibm-quantum-engagement/>) from the original on 6 March 2019. Retrieved 11 December 2018.
7. "Qiskit Github page" (<https://github.com/Qiskit>). *GitHub*. Archived (<https://web.archive.org/web/20210204053553/https://github.com/Qiskit>) from the original on 2021-02-04. Retrieved 2018-11-05.
8. Wille, R.; Meter, R. Van; Naveh, Y. (March 25, 2019). "IBM's Qiskit Tool Chain: Working with and Developing for Real Quantum Computers". *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (<https://ieeexplore.ieee.org/document/8715261>). pp. 1234–1240. doi:10.23919/DATE.2019.8715261 (<https://doi.org/10.23919%2FDATE.2019.8715261>). ISBN 978-3-9819263-2-3. S2CID 155108078 (<https://api.semanticscholar.org/CorpusID:155108078>). Archived (<https://web.archive.org/web/20221017103525/https://ieeexplore.ieee.org/document/8715261/>) from the original on 2022-10-17. Retrieved 2023-12-30.
9. "Qiskit in swift" (<https://github.com/Qiskit/qiskit-swift/>). *GitHub*. Archived (<https://web.archive.org/web/20211216230100/https://github.com/Qiskit/qiskit-swift/>) from the original on December 16, 2021. Retrieved September 24, 2019.
10. "Qiskit (Quantum Information Science Kit) for JavaScript" (<https://github.com/Qiskit/qiskit-js/>). *GitHub*. Archived (<https://web.archive.org/web/20180814153549/https://github.com/Qiskit/qiskit-js/>) from the original on August 14, 2018. Retrieved September 24, 2019.
11. "MicroQiskit" (<https://github.com/qiskit-community/MicroQiskit>). *GitHub*. Archived (<https://web.archive.org/web/20220212051406/https://github.com/qiskit-community/MicroQiskit>) from the original on February 12, 2022. Retrieved February 10, 2021.
12. "A collection of Jupyter notebooks showing how to use Qiskit that is synced with the IBM Quantum Experience" (<https://github.com/Qiskit/qiskit-tutorial/>). *GitHub*. Archived (<https://web.archive.org/web/20180707052518/https://github.com/QISKit/qiskit-tutorial/>) from the original on July 7, 2018. Retrieved September 24, 2019.
13. "Celebrating the IBM Q Experience community, and their research" (<https://www.ibm.com/blogs/research/2018/03/ibmq-published-research/>). *IBM*. IBM Research Editorial Staff. March 8, 2018. Archived (<https://web.archive.org/web/20191010224112/https://www.ibm.com/blogs/research/2018/03/ibmq-published-research/>) from the original on October 10, 2019. Retrieved September 24, 2019.
14. "Learn Quantum Computing using Qiskit" (<https://qiskit.org/textbook/preface.html>). Archived (<https://web.archive.org/web/20191220042259/https://qiskit.org/textbook/preface.html>) from the original on 20 December 2019. Retrieved 20 December 2019.
15. Javadi-Abhari, Ali; Gambetta, Jay M. (July 13, 2018). "Qiskit and its Fundamental Elements" (<https://medium.com/qiskit/qiskit-and-its-fundamental-elements-bcd7ead80492>). *Medium*. Archived (<https://web.archive.org/web/20190108055716/https://medium.com/qiskit/qiskit-and-its-fundamental-elements-bcd7ead80492>) from the original on 8 January 2019. Retrieved 10 January 2019.
16. "Qiskit Terra" (<https://web.archive.org/web/20191010224120/https://qiskit.org/Terra>). *Qiskit*. Archived from the original (<https://qiskit.org/Terra>) on October 10, 2019. Retrieved September 24, 2019.
17. "An Open High-Performance Simulator for Quantum Circuits" (<https://www.ibm.com/blogs/research/2018/05/quantum-circuits/>). *IBM*. IBM Research Editorial Staff. May 1, 2018. Archived (<https://web.archive.org/web/20190614124908/https://www.ibm.com/blogs/research/2018/05/quantum-circuits/>) from the original on June 14, 2019. Retrieved September 24, 2019.

18. Wood, Christopher J. (December 19, 2018). "Introducing Qiskit Aer: A high performance simulator framework for quantum circuits" (<https://medium.com/qiskit/qiskit-aer-d09d0fac7759>). *Medium*. Archived (<https://web.archive.org/web/20191101073346/https://medium.com/qiskit/qiskit-aer-d09d0fac7759>) from the original on November 1, 2019. Retrieved September 24, 2019.
19. "IBM Q Experience community publishes research" (<https://www.ibm.com/blogs/research/2018/03/ibmq-published-research/>). *IBM Research Blog*. 2018-03-08. Archived (<https://web.archive.org/web/20191010224112/https://www.ibm.com/blogs/research/2018/03/ibmq-published-research/>) from the original on 2019-10-10. Retrieved 2021-07-14.
20. Wootton, James R.; Loss, Daniel (2018). "Repetition code of 15 qubits". *Physical Review A*. **97** (5): 052313. arXiv:1709.00990 (<https://arxiv.org/abs/1709.00990>). Bibcode:2018PhRvA..97e2313W (<https://ui.adsabs.harvard.edu/abs/2018PhRvA..97e2313W>). doi:10.1103/PhysRevA.97.052313 (<https://doi.org/10.1103/PhysRevA.97.052313>). ISSN 2469-9926 (<https://www.worldcat.org/issn/2469-9926>). S2CID 53444623 (<https://api.semanticscholar.org/CorpusID:53444623>).
21. Roffe, Joschka; Headley, David; Chancellor, Nicholas; Horsman, Dominic; Kendon, Viv (2018). "Protecting quantum memories using coherent parity check codes". *Quantum Science and Technology*. **3** (3): 035010. arXiv:1709.01866 (<https://arxiv.org/abs/1709.01866>). Bibcode:2018QS&T....3c5010R (<https://ui.adsabs.harvard.edu/abs/2018QS&T....3c5010R>). doi:10.1088/2058-9565/aac64e (<https://doi.org/10.1088/2058-9565/aac64e>). ISSN 2058-9565 (<https://www.worldcat.org/issn/2058-9565>). S2CID 51684176 (<https://api.semanticscholar.org/CorpusID:51684176>).
22. Wang, Yuanhao; Li, Ying; Yin, Zhang-qi; Zeng, Bei (2018). "16-qubit IBM universal quantum computer can be fully entangled" (<https://doi.org/10.1038/s41534-018-0095-x>). *npj Quantum Information*. **4** (1): 46. arXiv:1801.03782 (<https://arxiv.org/abs/1801.03782>). Bibcode:2018npjQI...4...46W (<https://ui.adsabs.harvard.edu/abs/2018npjQI...4...46W>). doi:10.1038/s41534-018-0095-x (<https://doi.org/10.1038/s41534-018-0095-x>). ISSN 2056-6387 (<https://www.worldcat.org/issn/2056-6387>).
23. Zhukov, A. A.; Remizov, S. V.; Pogosov, W. V.; Lozovik, Yu. E. (2018). "Algorithmic simulation of far-from-equilibrium dynamics using quantum computer". *Quantum Information Processing*. **17** (9): 223. arXiv:1807.10149 (<https://arxiv.org/abs/1807.10149>). Bibcode:2018QuIP...17..223Z (<https://ui.adsabs.harvard.edu/abs/2018QuIP...17..223Z>). doi:10.1007/s11128-018-2002-y (<https://doi.org/10.1007/s11128-018-2002-y>). ISSN 1570-0755 (<https://www.worldcat.org/issn/1570-0755>). S2CID 254991672 (<https://api.semanticscholar.org/CorpusID:254991672>).
24. *Qiskit/qiskit-ignis* (<https://github.com/Qiskit/qiskit-ignis>), Qiskit, archived (<https://web.archive.org/web/20211221090311/https://github.com/Qiskit/qiskit-ignis>) from the original on 2021-12-21, retrieved 2021-12-21.
25. "Ignis provides tools for quantum hardware verification, noise characterization, and error correction" (<https://github.com/Qiskit/qiskit-ignis/>). *GitHub*. Archived (<https://web.archive.org/web/20190325181652/https://github.com/Qiskit/qiskit-ignis>) from the original on March 25, 2019. Retrieved September 24, 2019.
26. "Qiskit Algorithms Migration Guide – Qiskit 0.28.0 documentation" ([https://qiskit.org/documentation/aqua\\_tutorials/Qiskit%20Algorithms%20Migration%20Guide.html](https://qiskit.org/documentation/aqua_tutorials/Qiskit%20Algorithms%20Migration%20Guide.html)). *qiskit.org*. Archived ([https://web.archive.org/web/20210714012130/https://qiskit.org/documentation/aqua\\_tutorials/Qiskit%20Algorithms%20Migration%20Guide.html](https://web.archive.org/web/20210714012130/https://qiskit.org/documentation/aqua_tutorials/Qiskit%20Algorithms%20Migration%20Guide.html)) from the original on 2021-07-14. Retrieved 2021-07-14.
27. *Qiskit/qiskit-optimization* (<https://github.com/Qiskit/qiskit-optimization>), Qiskit, 2021-07-13, archived (<https://web.archive.org/web/20220121053415/https://github.com/Qiskit/qiskit-optimization>) from the original on 2022-01-21, retrieved 2021-07-14.

28. *Qiskit/qiskit-finance* (<https://github.com/Qiskit/qiskit-finance>), Qiskit, 2021-07-13, archived (<https://web.archive.org/web/20211218120510/https://github.com/Qiskit/qiskit-finance>) from the original on 2021-12-18, retrieved 2021-07-14.
29. *Qiskit/qiskit-machine-learning* (<https://github.com/Qiskit/qiskit-machine-learning>), Qiskit, 2021-07-13, archived (<https://web.archive.org/web/20210722092257/https://github.com/Qiskit/qiskit-machine-learning>) from the original on 2021-07-22, retrieved 2021-07-14.
30. *Qiskit/qiskit-nature* (<https://github.com/Qiskit/qiskit-nature>), Qiskit, 2021-07-13, archived (<https://web.archive.org/web/20210816151130/https://github.com/Qiskit/qiskit-nature>) from the original on 2021-08-16, retrieved 2021-07-14.

## Further reading

---

- [Qiskit Textbook \(https://qiskit.org/learn/\)](https://qiskit.org/learn/)
- 

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Qiskit&oldid=1198321194>"

▪