Introducing the IBM Quantum Experience

Quantum computing is here. Although today's quantum processors are modest in size, the complexity continues to grow every year. The time is ripe for us to build a community of new quantum learners and change the way we think about computing. The IBM Quantum team has put together this User Guide to help you understand the ***quantum world***. Our goal with the IBM Quantum Experience is for you to learn about quantum computing, compose your own experiments, run them in simulation, and even run them on the world's first fully-controllable quantum processor through the IBM Cloud.

The IBM Quantum Experience consists of

- a set of tutorials that will lead you from understanding the basics of simple single-qubit experiments (Section II (/qstage/#/tutorial?sectionId=71972f437b08e12d1f465a8857f4514c)) to more complicated multi-qubit experiments (Section III (/qstage/#/tutorial?sectionId=050edf961d485bfcd9962933ea09062b)), and then towards more advanced ideas in the area of quantum algorithms (Section IV (/qstage/#/tutorial?sectionId=8443c4f713521c10b1a56a533958286b)) and quantum error correction (Section V (/qstage/#/tutorial?sectionId=bfd2a30ad6c5c915da3a696d76c474d7)).
- the ***quantum Composer,*** which is a graphical user interface where you can compose your very own ***quantum score,*** much like a composer composes a music score.
- a simulator that can be used to test your quantum scores.
- access to a real live quantum processor running in one of the IBM Quantum Computing labs, where your quantum scores can be played.
- in the near future: a ***quantum community*** where your quantum scores, ideas, and experiences can be discussed and shared.

We do note that our IBM Quantum Experience is currently in a "Preview" phase as we engage with you the broader community to help refine its functionality and improve the overall interface. We have a bug tracker which you can access with the little bug icon at the bottom of each page. Please let us know your thoughts and opinions so that we can gather feedback to continually develop this experience, or please share with us any cool scores and results you might have observed!

To make sure we provide everyone a chance to use the real device, we have established a **Units** currency system. If you have joined the IBM Quantum Experience as a **Standard User,** you have full access to our simulation capabilities as well as previously run cached results from the device. When you work through and complete the tutorials, you will be rewarded with **Units** to run real-time experiments on the quantum processor hardware. This system allows us to manage queuing of the experiments that are run; at the moment we have just a single quantum processor connected to the cloud, and we'll have to properly time-allocate the instruction sets that are sent to it. Furthermore, you will notice that there will be intermittent periods when the real processor will be under maintenance and calibration. When your Units are used up, you can request for a replenish in the "Account" information page. There, you will notice also the chance for you to request an upgrade of your User status. Share with us your story, and tell us why you'd like to become an **Expert User** of the Quantum Experience.

For those who wish to jump to how the IBM Quantum Experience works, skip to the quantum Composer (/qstage/#/tutorial?sectionId=75a85f7e14ae3fd4329ad5c3e59466ea&pageIndex=3). If you wish to improve your understanding of the quantum world first, please continue reading the next few sections, where we will embark on a wild and fascinating journey that starts with a qubit.

We would like to acknowledge the work done under the IARPA Multi-Qubit Coherent Operations program as well as the LPS Quantum Characterization Validation & Verification program. The research performed in those programs have contributed to making this Quantum Experience possible.

Thank you,

Jay Gambetta, Jerry Chow, and the IBM Quantum team

The Quantum World

Today's computers perform calculations and process information using the standard (or as a physicist would say, "classical") model of computation, which dates back to Turing (https://en.wikipedia.org/wiki/Alan_Turing) and von Neumann (https://en.wikipedia.org/wiki/John_von_Neumann). In this model, all information is reducible to bits, which can take the values of either 0 or 1 -- and all processing can be performed via simple logic gates (https://en.wikipedia.org/wiki/Logic_gate) (AND, OR, NOT, NAND) acting on one or two bits at a time. At any point in its computation, a classical computer's state is entirely determined by the states of all its bits, so that a computer with n bits can exist in one of 2^n possible states, ranging from $00...0$ to $11\dots1$.

The power of the quantum computer, meanwhile, lies in its much richer repertoire of states. A quantum computer also has bits, just like any computer. But instead of 0 and 1, its quantum bits, or *qubits*, can represent a 0, 1, or both at once, a property known as superposition. This by itself is not much help, since a computer whose bits can be intermediate between 0 and 1 is just an analog computer, scarcely more powerful than an ordinary digital computer. A quantum computer takes advantage of a special kind of superposition that allows for **exponentially many** logical states at once, all the states from $|00...0\rangle$ to $|11\dots1\rangle$. This is a powerful feat, and no classical computer can achieve it. The vast majority of these quantum superpositions, and the ones most useful for quantum computation, are *entangled*—they are states of the whole computer that do not correspond to any assignment of digital or analog states of the individual qubits. While not as powerful as exponentially many classical computers, a quantum computer is significantly more powerful than any one classical computer -- whether it be deterministic, probabilistic, or analog. For a few famous problems (such as factoring large numbers), a quantum computer is clearly the winner over a classical computer. A working quantum computer could factor numbers in a day that would take a classical computer millions of years.

One might think that the difficulty in understanding quantum computing or quantum physics lies in "hard math"... but in fact, mathematically, quantum concepts are only a bit more complex than high school algebra. Quantum physics is hard because, like Einstein's theory of relativity, it requires internalizing ideas that are simple but very counterintuitive. With relativity, the strange idea is that time and space are interconnected, when common sense tells us they should act independently. If you try to explain relativity to someone by starting with time and space, you may be greeted by blank stares. A better way to start is as Einstein did, explaining that relativity follows from a simple physical principle: the speed of light is the same for all uniformly moving observers. This one modest idea then becomes extremely profound and leads, by inexorable logic, to Einsteinian spacetime.

With quantum physics, the counterintuitive ideas one must accept are 1) a physical system in a perfectly definite state can still behave randomly, and 2) two systems that are too far apart to influence each other can nevertheless behave in ways that, though individually random, are somehow strongly correlated. Unfortunately, unlike relativity, there is no single simple physical principle from which these conclusions follow. The best we can do is to distill quantum mechanics down to a few abstract-sounding mathematical laws, from which all the observed behavior of quantum particles (and qubits in a quantum computer) can be deduced and predicted. And, as with relativity, we must guard against attempting to describe quantum concepts in classical terms.

Quantum Laws in Black, White, and Blackandwhite

Quantum laws are, as far as we know, the most fundamental physical laws; they are inviolable. Here is our version of quantum physics distilled to five key laws**.**

Quantum is a system like everything else.

To each physical system there corresponds a Hilbert space (1) of dimensionality equal to the system's maximum number of reliably distinguishable states (2).

A quantum state is a configuration of the system.

Each direction (ray) in the Hilbert space corresponds to a possible state of the system (3), with two states being reliably distinguishable if and only if their directions are orthogonal (inner product is zero).

A quantum state changes; it naturally wants to evolve, but it can always be *undone*.

Evolution of a closed system is a unitary (4) transformation on its Hilbert space.

Scaling - how parts make a whole.

The Hilbert space of a composite system is the tensor product of the Hilbert space of the parts (5).

Quantum measurements are probabilistic.

Each possible measurement (6) on a system corresponds to a resolution of its Hilbert space into orthogonal subspaces $\{\Pi_j\}$ where $\sum_j \Pi_j = 1$. On state $|\psi\rangle$ the result j occurs with probability $P(j) = \langle\psi|\Pi_j|\psi\rangle$ and the state after the measurement is $|\psi_j\rangle = \Pi_j|\psi\rangle/\sqrt{P(j)}$.

These five principles are the foundation for the whole quantum world.

Clarifications

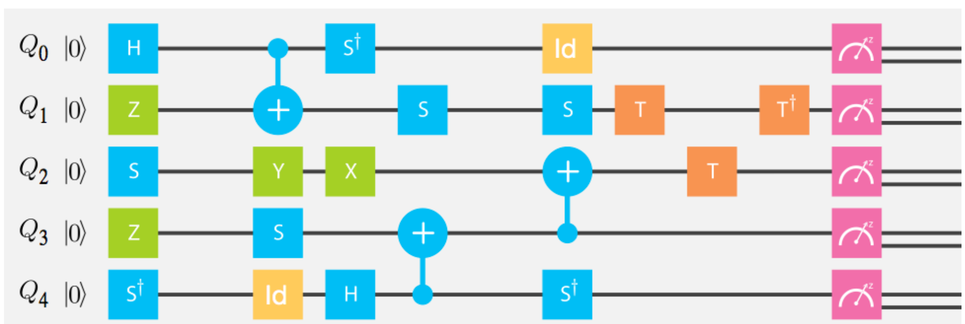
1. A Hilbert space is a linear vector space with complex coefficients and inner products $\langle\phi|\psi\rangle = \sum_i \phi_i^* \psi_i$.
2. For a single qubit, there are two standard orthogonal states (*computational basis states*) that are conventionally denoted $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.
3. Other qubit states include $|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $|-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $|\circlearrowright\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ i \end{pmatrix}$ and $|\circlearrowleft\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -i \end{pmatrix}$
4. *Unitary* means linear and inner product preserving.
5. A two-qubit system can exist in a product state such as $|00\rangle$ or $|0+\rangle$ but also in an *entangled* state $(|00\rangle + |11\rangle)/\sqrt{2}$, in which neither qubit has a definite state, even though the pair together does.
6. Measurement causes the system to behave probabilistically and forget its pre-measurement state, unless that state happens to lie entirely with one of the subspaces Π_j.

The Quantum Composer

The ***Quantum Composer*** is our graphical user interface for programming a quantum processor. Those familiar with quantum computing may recognize the composer as a tool to construct *quantum circuits* using a library of well-defined gates and measurements. For those not familiar, we will explain a few of the key parts.

When you first click on the "Composer" tab above, you will be asked to determine whether you would like to run an *ideal* quantum processor or a *real* quantum processor. This refers to the topology of the system. In the ideal processor, gates can be placed anywhere, whereas in the real processor, the topology is set by the physical device that is currently running in our lab (note that this restricts the usability of some of the two-qubit gates).

Once you are in the "Composer" tab, you can start making your very own quantum circuits!



We call this depiction a *quantum score* because it resembles a musical score in several respects. Time progresses from left to right. Each line represents a qubit (as well as what happens to that qubit over time). Each qubit has a different frequency, like a different musical note. Quantum gates are represented by square boxes that play a frequency for different durations, amplitudes, and phases. These are called single-qubit gates. The gates made with vertical lines that connect two qubits together are known as CNOT gates; these two-qubit gates function like an exclusive OR gate in conventional digital logic. The qubit at the solid dot end of the CNOT gate controls the inversion of the state of the qubit at the ⊕ end of the gate (hence controlled NOT, or CNOT). Some gates, like the CNOT, have hardware constraints; the set of allowed connections is defined in the schematic of the device located below the Quantum Composer, along with recently calibrated device parameters.

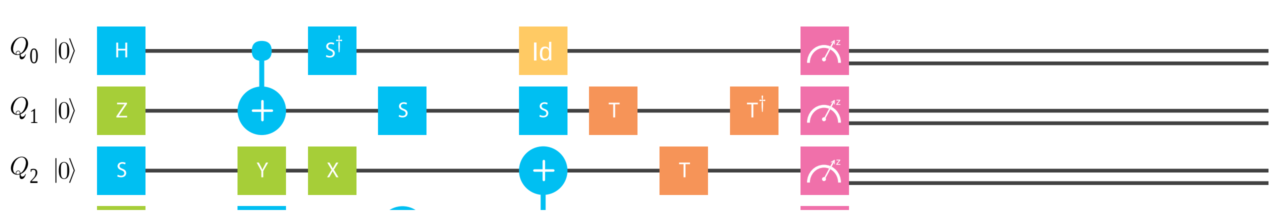
The Quantum Composer's library (located below the qubit stave) contains four classes of gates, each denoted by its own color. Hitting the help button on the right bar gives a quick summary of all the different gates. The first class of gates (yellow) represents an idle operation on the qubit for a time equal to the single-qubit gate duration. The second class of gates (green) represents a group known as *Pauli operators*, which represent bit-flips (X, which is a classical NOT), phase-flips (Z), and a combined bit-flip and phase-flip (Y). The third class (blue) represents *Clifford* gates, which consist of H, S, and $S^\dagger$ gates for generating quantum superpositions, along with the all-important CNOT two-qubit gate which is necessary for entanglement. The final class (orange) represents gates that are required for universal control.

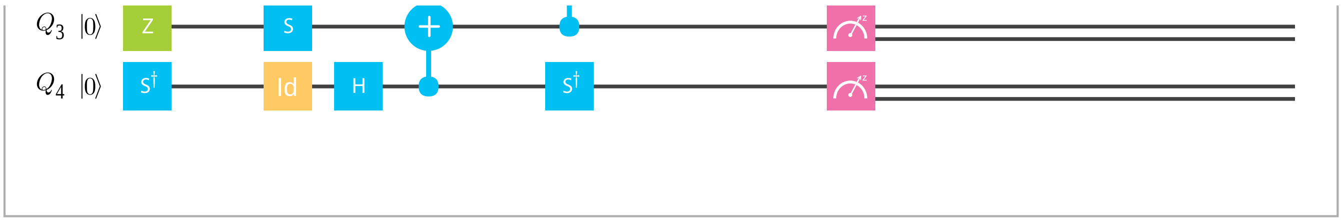
A quantum algorithm (circuit) begins by preparing the qubits in well-defined states (here the ground state, $|0\rangle$, which we've automatically done for you), then executing a series of one- and two-qubit gates in time, followed by a measurement of the qubits. Measurements are depicted with pink boxes. There are two options for measurement: a *standard measurement* which is a simple Z projection, or a *Bloch measurement* which is a Bloch sphere projection to indicate the final qubit state having been projected along the X, Y and Z axes. After measurement, the qubit state becomes classical (either $|0\rangle$ or $|1\rangle$ but never a quantum superposition); this is represented by the double line that appears after a measurement operation. All these elements will become clearer as you walk through this user guide.

To use the Composer, simply drag the gate boxes into the qubit stave to place them. Double-tap the boxes to delete, or drag them to the trash bin. For the CNOT gate, drag into the stave and place first to indicate the target qubit, and then click again on the control qubit. Note that once you place a measurement, you cannot follow it with subsequent gates, because the information at that point has become classical.

Load the quantum score and try out simulating it to see what it does, or start composing your own!

Random Score





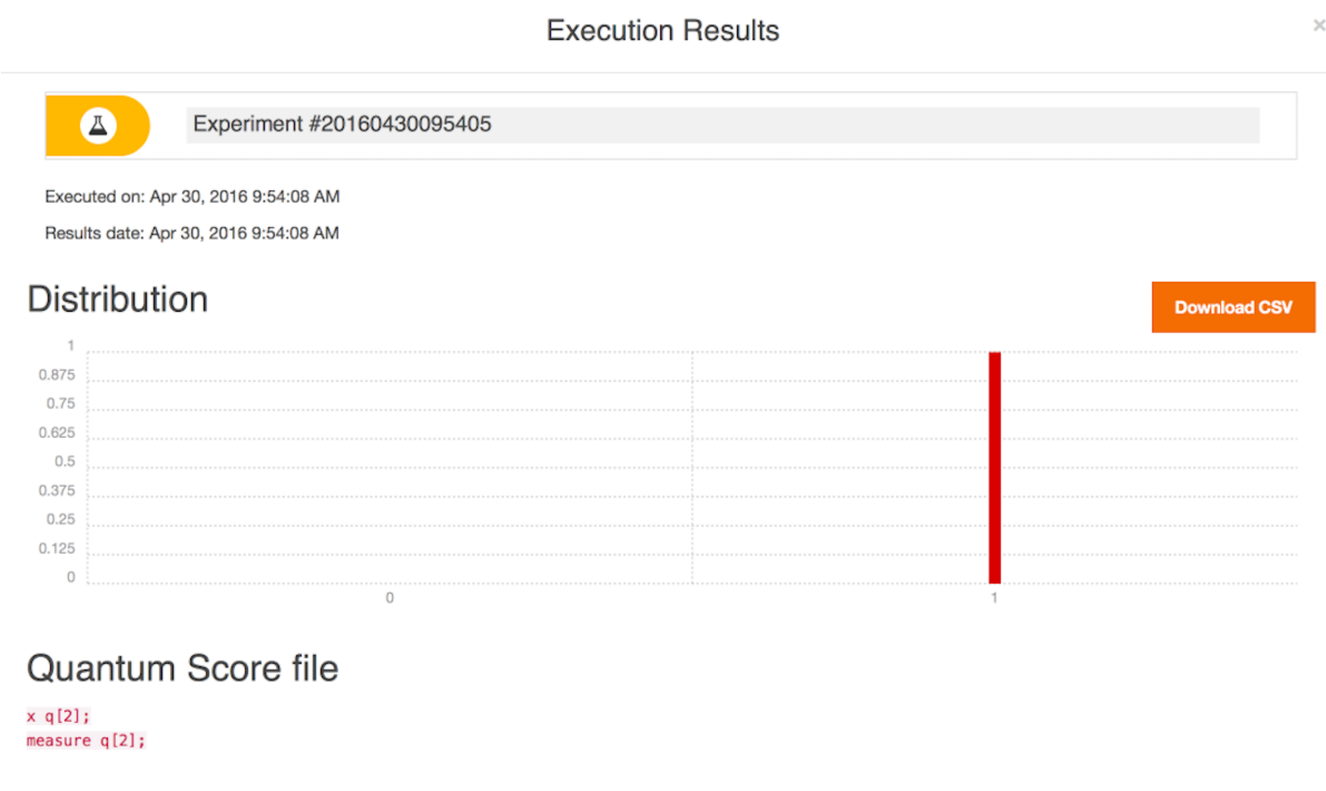
Q_3 $|0\rangle$
Q_4 $|0\rangle$
Z
S
S†
Id
H
S†

Running your Quantum Scores

So now that we've gotten familiar with the Composer, how do we run it? When you begin an experiment, you'll be prompted to give it a name, which is helpful if you'd like to quickly recognize it, or go back to it for reference.

Running on an Ideal Quantum Processor

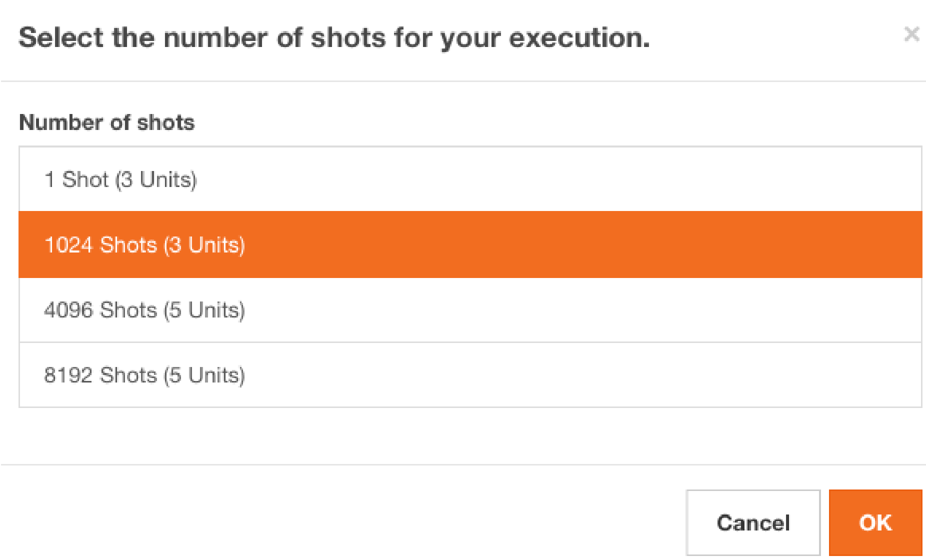
If you've selected an "Ideal Quantum Processor," your only option is to run your score in simulation. That is because the ideal processor permits more connections than are available with the real device, which is limited by physical connectivity. If you "Simulate," your execution happens immediately, and your results are returned to you as shown below.



Running on a Real Quantum Processor with Units

If you instead select a "Real Quantum Processor," you can populate your qubit stave with gates and measurements, then hit "Run" to launch your score on the actual quantum computer. However, this will now require the usage of Units, a pseudo-currency we've devised to help us manage the queuing of instructions to our quantum processor. Having signed up as a Standard User of the IBM Quantum Experience, you do not have any Units yet, but slots for up to 10 Units. Work your way through the tutorials in the User Guide to fill up those 10 Units!

Now when you have Units, upon hitting "Run" a new box will show on your screen asking how many "shots" or repeats of your score you would like to run, along with the associated cost in Units.



After hitting "Ok," you are then given the chance to either run a brand new execution and spend the necessary Units, or to immediately receive results from a previous run of the same score, from our cache, for free. You will notice that many of the tutorial executions have a number of previous executions that you can observe and learn from!

Running a brand new execution places your score into the experimental queue, and you will be notified via e-mail when your score has been run. As a Standard User, your Units can be replenished by request from the "Account" at the top of your screen.

Upon running your score, its progress will be visible in the "My Scores" tab, sorted by date executed. When your results are ready, you will be able to view them from this tab. You can also re-edit your score and launch it into a simulator while you wait for your results to return.