

# A Practical Introduction to Quantum Computing: From Qubits to Quantum Machine Learning and Beyond

Elías F. Combarro  
[combarro@gmail.com](mailto:combarro@gmail.com)

CERN openlab (Geneva, Switzerland) - University of Oviedo (Oviedo, Spain)

CERN - November/December 2020



Universidad de Oviedo

## Part I

Introduction: quantum computing...  
the end of the world as we know it?

# I, for one, welcome our new quantum overlords

NEWS

QUANTUM PHYSICS

## Google officially lays claim to quantum supremacy

A quantum computer reportedly beat the most powerful supercomputers at one type of calculation

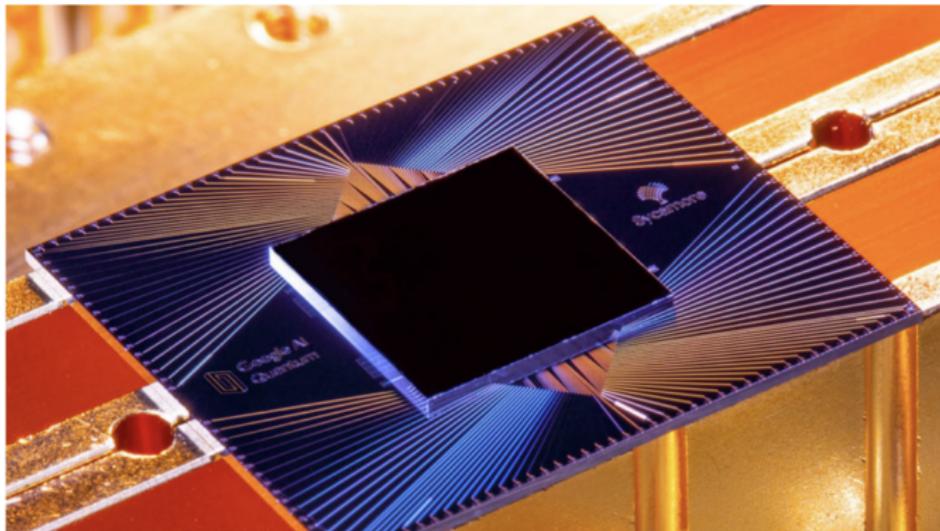


Image credits: [sciencenews.org](http://sciencenews.org)

## Philosophy of the course

If you can't  
explain it to a  
**computer**  
you don't  
understand it  
yourself.

ALBERT EINSTEIN

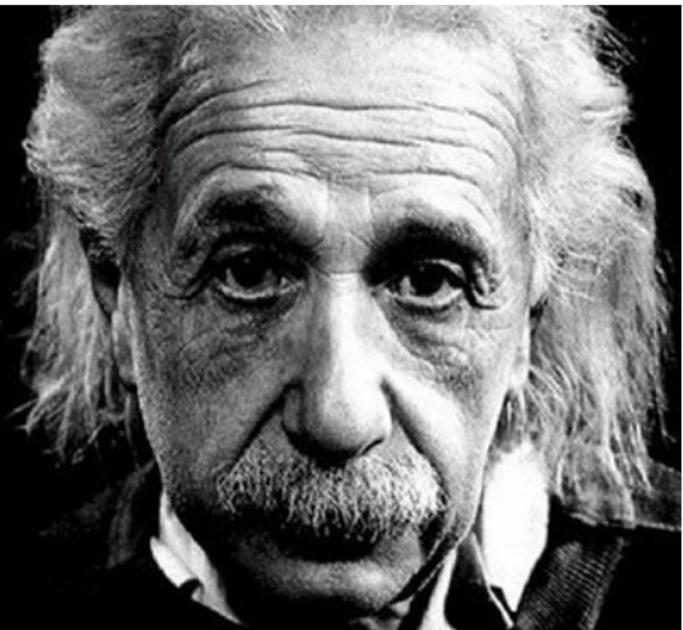


Image credits: Modified from an Instagram image by Bob MacGuffie

# Tools and resources

- Jupyter Notebooks
  - Web application to create and execute notebooks that include code, images, text and formulas
  - They can be used locally (Anaconda) or in the cloud (mybinder.org, Google Colab...)
- IBM Quantum Experience
  - Free online access to quantum simulators (up to 32 qubits) and **actual quantum computers** (1, 5 and 15 qubits) with different topologies
  - Programmable with a visual interface and via different languages (python, qasm, Jupyter Notebooks)
  - Launched in May 2016
  - <https://quantum-computing.ibm.com/>

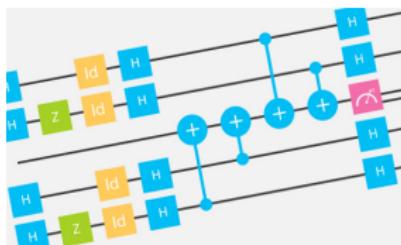


Image credits: IBM

## Tools and resources (2)

- Quirk
  - Online simulator (up to 16 qubits)
  - Lots of different gates and visualization options
  - <http://algassert.com/quirk>
- D-Wave Leap
  - Access to D-Wave quantum computers
  - Ocean: python library for quantum annealing
  - Problem specific (QUBO, Ising model...)
  - <https://www.dwavesys.com/take-leap>



# The shape of things to come



Image credits: Created with wordclouds.com

# What is quantum computing?

## Quantum computing

Quantum computing is a computing paradigm that exploits quantum mechanical properties (superposition, entanglement, interference...) of matter in order to do calculations

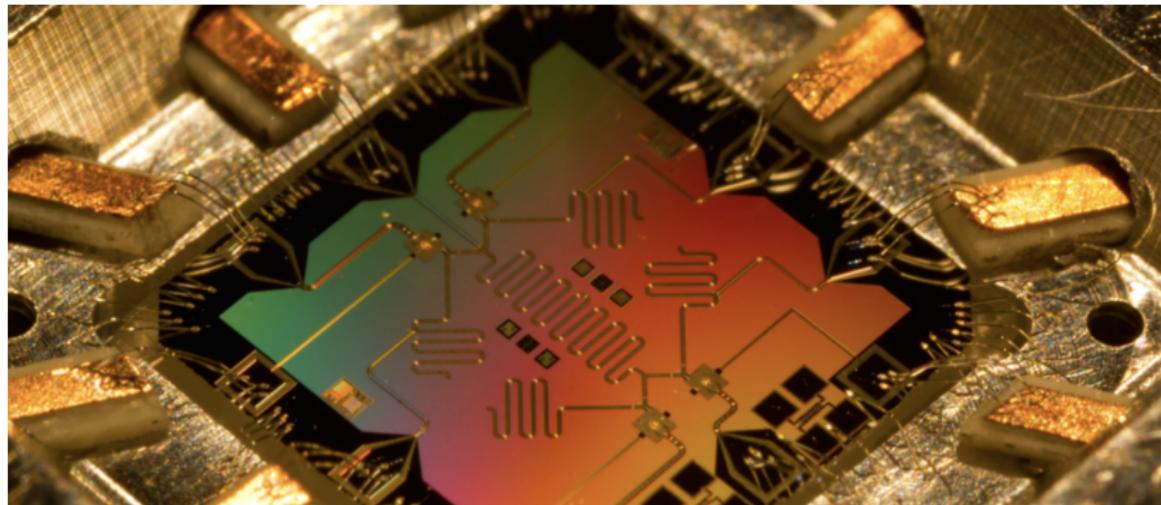


Image credits: Erik Lucero

# Models of quantum computing

- There are several models of quantum computing (they're all equivalent)
  - Quantum Turing machines
  - **Quantum circuits**
  - Measurement based quantum computing (MBQC)
  - Adiabatic quantum computing
  - Topological quantum computing
- Regarding their **computational capabilities**, they are equivalent to classical models (Turing machines)

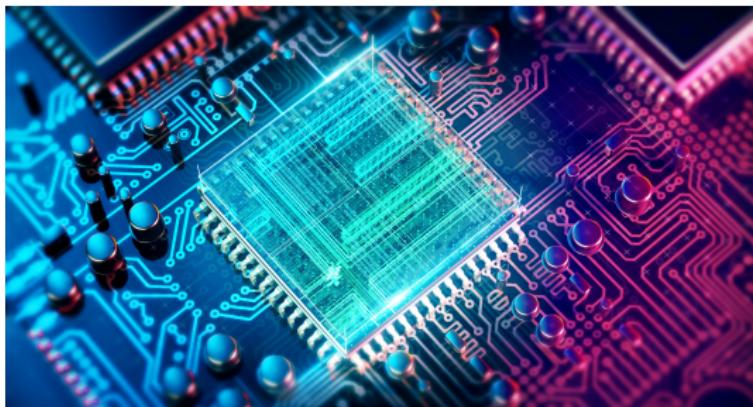


Image credits: Getty Images

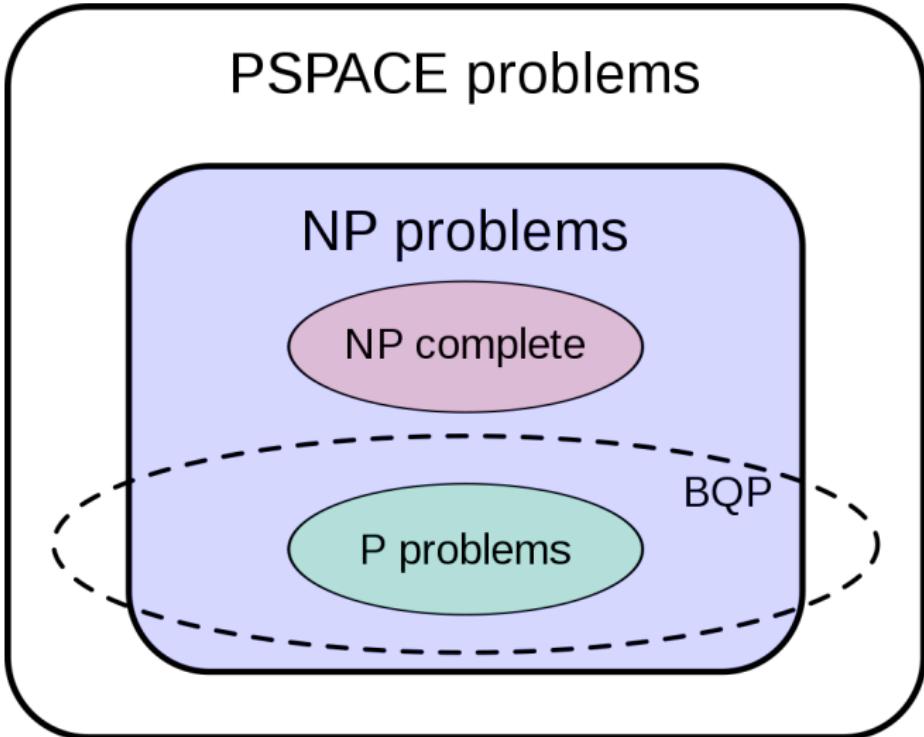
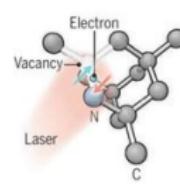
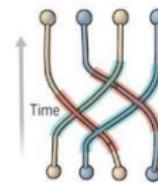
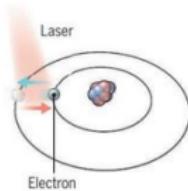
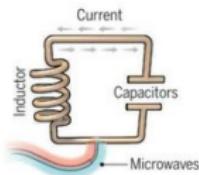


Image credits: wikipedia.org

# What technologies are used to build quantum computers?



## Superconducting loops

### Company support

Google, IBM, Quantum Circuits

### Pros

Fast working. Build on existing semiconductor industry.

### Cons

Collapse easily and must be kept cold.

## Trapped ions

ionQ

## Silicon quantum dots

Intel

## Topological qubits

Microsoft,  
Bell Labs

## Diamond vacancies

Quantum Diamond  
Technologies

Image credits: Graphic by C. Bickle/Science data by Gabriel Popkin

# What is a quantum computer like?



Image credits: IBM

The Sounds of IBM: IBM Q

# Programming a quantum computer

- Different frameworks and programming languages:
  - qasm
  - Qiskit (IBM)
  - Cirq (Google)
  - Forest/pyqil (Rigetti)
  - Q# (Microsoft)
  - Ocean (D-Wave)
  - ...
- Most of them for quantum circuit specification

Switch to Composer

Backend: Custom Topology Experiment Units: 3

Simulate

The screenshot shows the IBM Qiskit interface. On the left, there is a code editor window titled "Switch to Composer" containing QASM code. On the right, there is a graphical interface for creating quantum circuits. The circuit has three qubits (q[0], q[1], q[2]) and three classical bits (c[0], c[1], c[2]). The circuit consists of several operations: initializations, a sequence of gates (including H, CNOT, and T), and measurements. The graphical interface includes controls for "Simulate" and "Backend: Custom Topology".

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
creg c[1];
creg c1[1];
creg c2[1];
gate post q { }
u3(0.3,0.2,0.1) q[0];
h q[1];
cx q[1],q[2];
barrier q[1];
cx q[0],q[1];
h q[0];
measure q[0] -> c0[0];
measure q[1] -> c1[0];
if(c0==1) x q[2];
if(c1==1) x q[2];
post q[2];
measure q[2] -> c2[0];

```

Import QASM      Download QASM

Image credits: IBM

# What are the elements of a quantum circuit?

- Every computation has three elements: data, operations and results
- In quantum circuits:
  - Data = **qubits**
  - Operations = **quantum gates** (unitary transformations)
  - Results = **measurements**



Image credits: Adobe Stock

## Part II

One-qubit systems: one qubit to rule them all

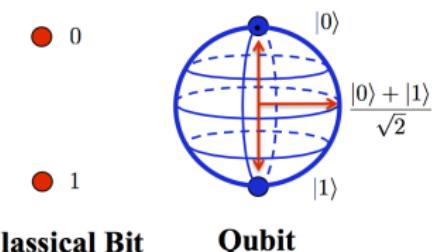
# What is a qubit?

- A classical bit can take two different values (0 or 1). It is discrete.
- A qubit can “take” **infinitely** many different values. It is continuous.
- Qubits live in a **Hilbert vector space** with a basis of two elements that we denote  $|0\rangle$  y  $|1\rangle$ .
- A generic qubit is in a **superposition**

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where  $\alpha$  and  $\beta$  are **complex numbers** such that

$$|\alpha|^2 + |\beta|^2 = 1$$



# Measuring a qubit

- The way to know the value of a qubit is to perform a measurement. However
  - The result of the measurement is random
  - When we measure, we only obtain one (classical) bit of information
- If we measure the state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  we get 0 with probability  $|\alpha|^2$  and 1 with probability  $|\beta|^2$ .
- Moreover, the new state after the measurement will be  $|0\rangle$  or  $|1\rangle$  depending of the result we have obtained (wavefunction collapse)
- We cannot perform several independent measurements of  $|\psi\rangle$  because we cannot copy the state (**no-cloning theorem**)



# The Bloch sphere

- A common way of representing the state of a qubit is by means of a point in the surface of the Bloch sphere
- If  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  with  $|\alpha|^2 + |\beta|^2 = 1$  we can find angles  $\gamma, \delta, \theta$  such that

$$\alpha = e^{i\gamma} \cos \frac{\theta}{2}$$

$$\beta = e^{i\delta} \sin \frac{\theta}{2}$$

- Since an overall phase is physically irrelevant, we can rewrite

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

with  $0 \leq \theta \leq \pi$  and  $0 \leq \varphi < 2\pi$ .

## The Bloch sphere (2)

- From  $|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$  we can obtain spherical coordinates for a point in  $\mathbb{R}^3$   
 $(\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta)$

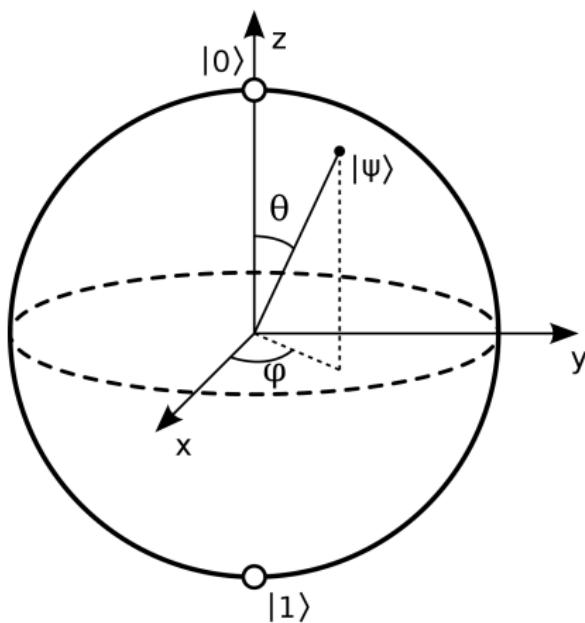


Image credits: wikipedia.org

# What are quantum gates?

- Quantum mechanics tells us that the evolution of an isolated state is given by the Schrödinger equation

$$H(t)|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle$$

- In the case of quantum circuits, this implies that the operations that can be carried out are given by unitary matrices. That is, matrices  $U$  of complex numbers verifying

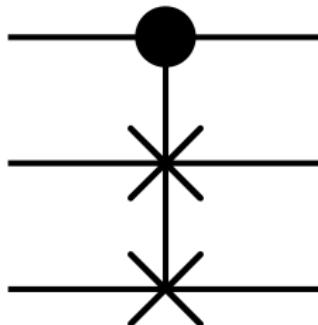
$$UU^\dagger = U^\dagger U = I$$

where  $U^\dagger$  is the conjugate transpose of  $U$ .

- Each such matrix is a possible quantum gate in a quantum circuit

# Reversible computation

- As a consequence, all the operations have an inverse:  
**reversible computing**
- Every gate has the same number of inputs and outputs
- We cannot directly implement some classical gates such as *or*, *and*, *nand*, *xor*...
- But we can simulate any classical computation with small overhead
- Theoretically, we could compute without wasting energy  
(Landauer's principle, 1961)



# One-qubit gates

- When we have just one qubit  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , we usually represent it as a column vector  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
- Then, a one-qubit gate can be identified with a matrix  $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  that satisfies

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

where  $\bar{a}, \bar{b}, \bar{c}, \bar{d}$  are the conjugates of complex numbers  $a, b, c, d$ .

## Action of a one-qubit gate

- A state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  is transformed into

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} a\alpha + b\beta \\ c\alpha + d\beta \end{pmatrix}$$

that is, into the state  $|\psi\rangle = (a\alpha + b\beta)|0\rangle + (c\alpha + d\beta)|1\rangle$

- Since  $U$  is unitary, it holds that

$$|(a\alpha + b\beta)|^2 + |(c\alpha + d\beta)|^2 = 1$$

# The $X$ or $NOT$ gate

- The  $X$  gate is defined by the (unitary) matrix

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

- Its action (in quantum circuit notation) is

$$|0\rangle \xrightarrow{\boxed{X}} |1\rangle$$

$$|1\rangle \xrightarrow{\boxed{X}} |0\rangle$$

that is, it acts like the classical  $NOT$  gate

- On a general qubit its action is

$$\alpha |0\rangle + \beta |1\rangle \xrightarrow{\boxed{X}} \beta |0\rangle + \alpha |1\rangle$$

# The $Z$ gate

- The  $Z$  gate is defined by the (unitary) matrix

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

- Its action is

$$|0\rangle \xrightarrow{\boxed{Z}} |0\rangle$$

$$|1\rangle \xrightarrow{\boxed{Z}} -|1\rangle$$

# The $H$ or Hadamard gate

- The  $H$  or Hadamard gate is defined by the (unitary) matrix

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- Its action is

$$|0\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|1\rangle \xrightarrow{H} \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

- We usually denote

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

and

$$|-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

# Other important gates

- $Y$  gate

$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

- $S$  gate

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$$

- $T$  gate

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

- The gates  $X$ ,  $Y$  and  $Z$  are also called, together with the identity, the Pauli gates. An alternative notation is  $\sigma_X$ ,  $\sigma_Y$ ,  $\sigma_Z$ .

# Rotation gates

- We can define the following rotation gates

$$R_X(\theta) = e^{-i\frac{\theta}{2}X} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}X = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_Y(\theta) = e^{-i\frac{\theta}{2}Y} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}Y = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}Z} = \cos \frac{\theta}{2}I - i \sin \frac{\theta}{2}Z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

- Notice that  $R_X(\pi) \equiv X$ ,  $R_Y(\pi) \equiv Y$ ,  $R_Z(\pi) \equiv Z$ ,  
 $R_Z(\frac{\pi}{2}) \equiv S$ ,  $R_Z(\frac{\pi}{4}) \equiv T$

## Using rotation gates to generate one-qubit gates

- For any one-qubit gate  $U$  there exist a unit vector  $r = (r_x, r_y, r_z)$  and an angle  $\theta$  such that

$$U \equiv e^{-i\frac{\theta}{2}r \cdot \sigma} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} (r_x X + r_y Y + r_z Z)$$

- For instance, choosing  $\theta = \pi$  and  $r = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$  we can see that

$$H \equiv e^{-i\frac{\theta}{2}r \cdot \sigma} = -i \frac{1}{\sqrt{2}} (X + Z)$$

- Additionally, it can also be proved that there exist angles  $\alpha$ ,  $\beta$  and  $\gamma$  such that

$$U \equiv R_Z(\alpha)R_Y(\beta)R_Z(\gamma)$$

# Hello, quantum world!

- Our very first quantum circuit!



- After applying the  $H$  gate the qubit state is

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

- When we measure, we obtain 0 or 1, each with 50% probability: we have a circuit that generates perfectly uniform random bits!