

Ch4: Programming a Quantum Algorithm with Qiskit



Python programming

Qiskit is a quantum programming library built upon Python 3.



Why Python ?

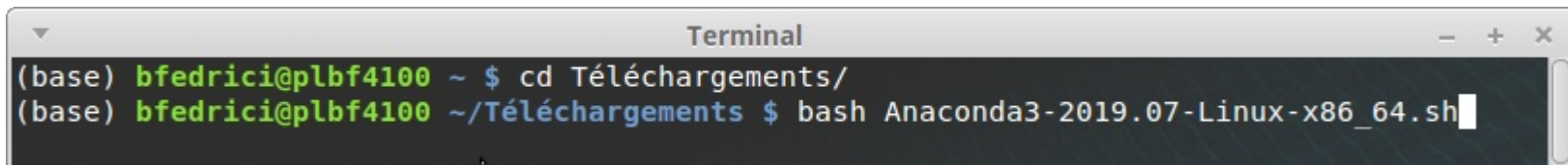
- Library ecosystem
- Low entry barrier
- Interpreted
- ...

Anaconda distribution

We recommend working with Python 3 Anaconda distribution as it comes with more than 1,500 packages as well as the conda package manager and virtual environment manager.

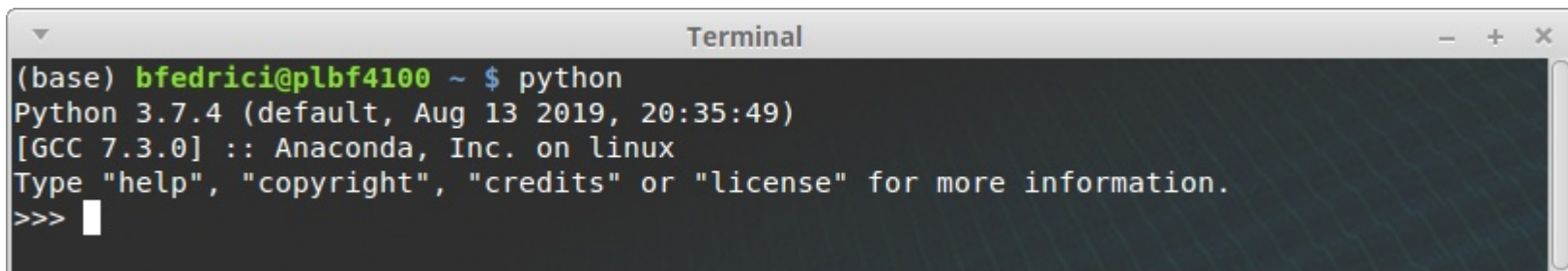
Step 1 – Anaconda Download (Python v3.7): <https://www.anaconda.com/distribution/>

Step 2 – Anaconda Installation Command (from Linux Terminal):



```
Terminal
(base) bfredrici@plbf4100 ~ $ cd Téléchargements/
(base) bfredrici@plbf4100 ~/Téléchargements $ bash Anaconda3-2019.07-Linux-x86_64.sh
```

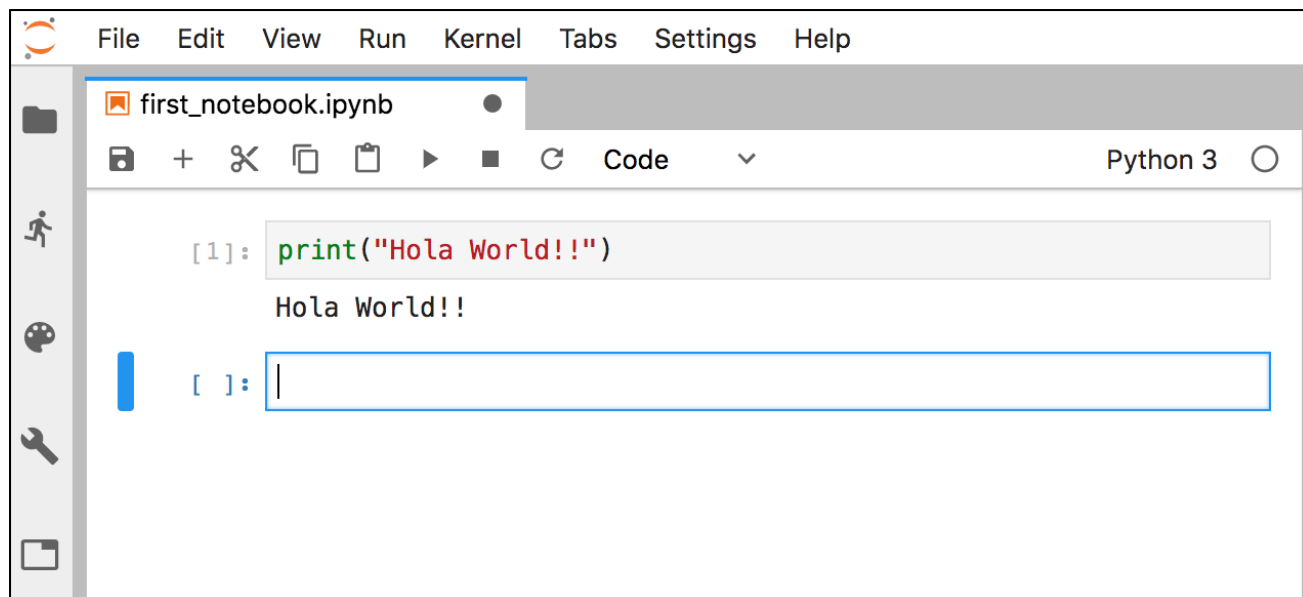
Step 3 – Check Installation (from Linux Terminal):



```
Terminal
(base) bfredrici@plbf4100 ~ $ python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Jupyter notebooks

One way to code in Python is to use a Jupyter (IPython) notebook. The free Jupyter notebook environment will make our data analysis easier to record, understand and reproduce. Jupyter notebook environment is available by default in Anaconda.





Qiskit

Qiskit is an open-source framework for quantum computing. It provides tools for creating and manipulating quantum programs and running them on prototype quantum devices on IBM Q Experience or on simulators – classical devices.

```
In [7]: from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
        from qiskit.tools.visualization import circuit_drawer
        import numpy as np

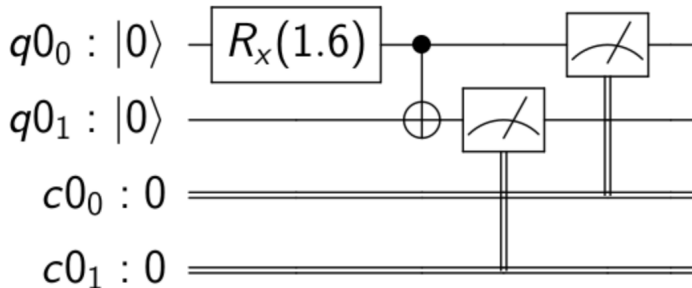
        qr = QuantumRegister(2)
        cr = ClassicalRegister(2)
        qp = QuantumCircuit(qr, cr)

        qp.rx( np.pi/2, qr[0])
        qp.cx(qr[0], qr[1])

        qp.measure(qr, cr)

        circuit_drawer(qp)
```

Out[7]:



- Qiskit documentation:
<https://qiskit.org/documentation/>
- Qiskit repositories:
<https://github.com/Qiskit>
- Qiskit channel:
<https://www.youtube.com/Qiskit>



Installation

Qiskit is an open-source framework for quantum computing. It provides tools for creating and manipulating quantum programs and running them on prototype quantum devices on IBM Q Experience or on simulators – classical devices.

Step 1 – Qiskit Installation Command (from Linux Terminal):



```
Terminal
(base) bfredrici@plbf4100 ~ $ pip install qiskit
```


Step 2 – Check Installation (from Jupyter notebook):

```
Entrée [1]: import qiskit
```

```
Entrée [2]: qiskit.__qiskit_version__
```

```
Out[2]: {'qiskit-terra': '0.10.0',
        'qiskit-aer': '0.3.2',
        'qiskit-ignis': '0.2.0',
        'qiskit-ibmq-provider': '0.3.3',
        'qiskit-aqua': '0.6.1',
        'qiskit': '0.13.0'}
```

Qiskit libraries

 Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

[Sign in](#)

[Sign up](#)



Qiskit

An open-source framework for working with noisy quantum computers at the level of pulses, circuits, and algorithms.

<https://qiskit.org> [✉ qiskit@qiskit.org](mailto:qiskit@qiskit.org)

 **Repositories** 22

 Packages

 People 37

**qiskit**

Qiskit is an open-source framework for working with noisy quantum computers at the level of pulses, circuits, and algorithms.

Python ★ 738 🍴 295

**qiskit-ixx-tutorials**

A collection of Jupyter notebooks showing how to use Qiskit that is synced with the IBM Q Experience

Jupyter Notebook ★ 1.1k 🍴 663

**qiskit-terra**

Terra provides the foundations for Qiskit. It allows the user to write quantum circuits easily, and takes care of the constraints of real hardware.

Python ★ 2.7k 🍴 964

**qiskit-aer**

Aer is a high performance simulator for quantum circuits that includes noise models

Python ★ 116 🍴 134

**qiskit-aqua**

Quantum Algorithms & Applications in Python.

Python ★ 289 🍴 215

**qiskit-ignis**

Ignis provides tools for quantum hardware verification, noise characterization, and error correction.

Python ★ 68 🍴 78

Type: All ▾

Language: All ▾