Anaconda/Python Installation

What is Anaconda/Python?

Anaconda is **a distribution of the Python and R programming languages** for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

Python is a high level programming language. It was created by Guido van Rossum, and released in 1991.

The main **difference between Anaconda** and **Python is**, **Anaconda is** a distribution of **Python** and R **programming** languages for data science and Machine learning tasks whereas **Python is** a high-level general-purpose **programming** language. The package manager in **Anaconda is** called Conda while for the **Python** it **is** pip

It is used for:

- web development (server-side),
- software development,
- mathematics,
- Scientific communities and many more

- To install Anaconda Python follow the instructions at Anaconda Distribution Website. Based on the operating system, select the proper version of the Anaconda package and install it on your PC.

- After you successfully install the proper version, you will get an anaconda application on your PC.

Jupyter notebook

Jupyter is a free, open-source, interactive web tool known as a computational **notebook**, which researchers can **use** to combine software code, computational output, explanatory text and multimedia resources in a single document. The ***Jupyter Notebook*** can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

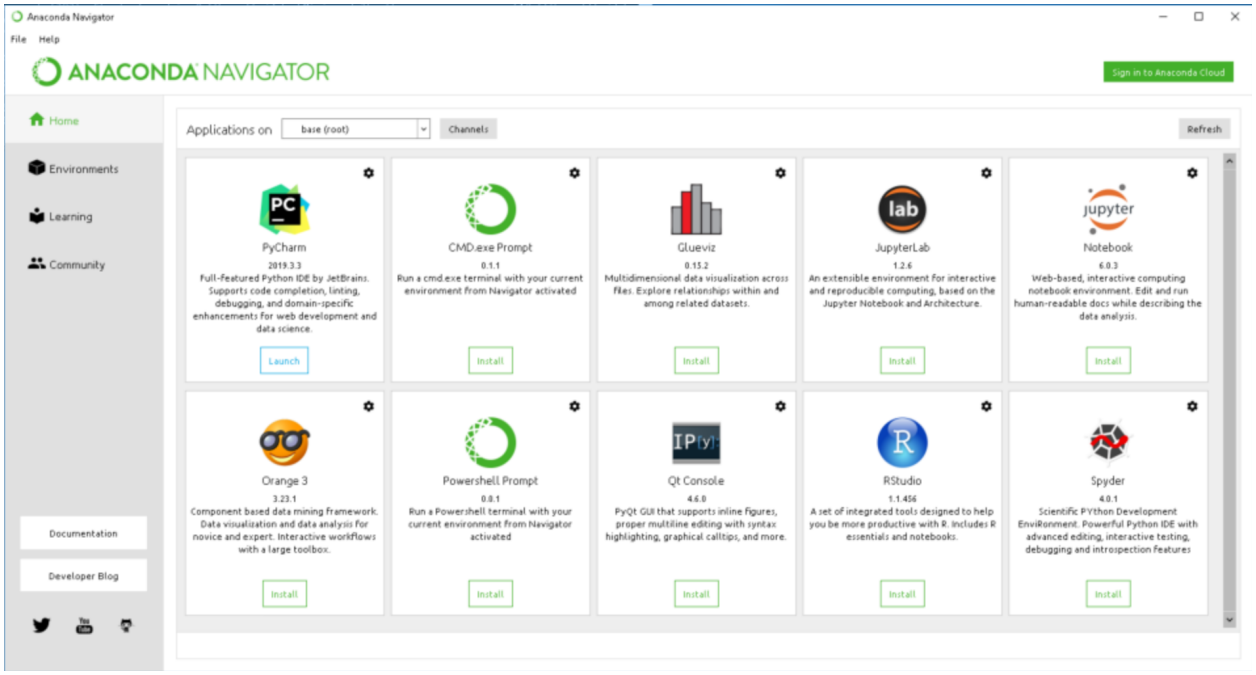
- By far the easiest way to install Jupyter is with Anaconda. Jupyter notebooks automatically come with the distribution. You'll be able to use notebooks from the default environment.

To install Jupyter notebooks in a conda environment, use

conda install jupyter notebook

Jupyter notebooks are also available through pip with

pip install jupyter notebook



The best way to start with this the "Jupyter notebook" is to Lunch the jupyter notebook from anaconda navigator and start the coding session.

Qiskit Installation

What is Qiskit

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

Install Locally

Qiskit has a Python module/package that allows us to create quantum circuits, run them, and work with different quantum algorithms. Qiskit package has a built-in quantum simulator to run quantum circuits locally and allows integration with IBM Quantum to run circuits or algorithms on real quantum computers.

- Qiskit supports Python 3.6 or later.(recommend installing Anaconda)
- Install the Qiskit package
  [Optional: you can create a new virtual python environment and work on it
  conda create -n ENV_NAME
  Activate the virtual environment
  conda activate ENV_NAME
  Install QISKIT package
  pip install qiskit
- [Optional] For better visualization in jupyter notebook, install qiskit with visualization
  pip install qiskit[visualization]

On Cloud

We can use few different platforms to use qiskit. An advantage of running on the cloud is that the load on your computer is less and few cloud services are comparatively faster than the average desktop. And the disadvantage is initiation might take some time, it might be a bit slower.

IBM Quantum lab

- Create an account on IBM Quantum Lab
- Launch IBM Quantum Lab
- Create a new notebook and rename it

It has qiskit preinstalled into it.

Google Colab, Kaggle, Strangework

You need to install the qiskit package every time you launch Google Colab and Kaggle. However, you do not need to install the qiskit yourself at Strangework.

Qiskit WorkFlow

The basic workflow of qiskit is as follows

- Build: Design a quantum circuit(s) for your problem
- Compile: Compile circuits for a specific quantum service, e.g. a quantum system or classical simulator.
- Run: Run the compiled circuits on the quantum system or simulator. These services can be cloud-based or local.
- Analyze: Use the result from the experiment and analyze them using techniques like visualization

Elements of Qiskit

Qiskit has four main elements

- Terra: This is the foundation of qiskit, acting as the bedrock for composing quantum programs. It works at the level of gates circuits and pulses. We use this to build circuits.

- Aer: “Aer, the ‘air’ element, permeates all Qiskit elements.” It contains simulators, emulators. We can use various elements of Aer to verify the possible prospects of an algorithm for future use.

- Ignis: Ignis is the ‘fire’ element. It works to reduce noise and errors within the experiments. It includes error correction codes, noise mitigation, and other protective layers.
- Aqua: Aqua is the ‘water’ element in Qiskit. As water is the element of life, Aqua takes its purpose as the element to work with real-world applications. “Aqua is where algorithms for quantum computers are built.” Aqua is being used by experts n different fields like chemistry, optimization, finance, and AI.