QUANTUM KICKSTART · BEFORE SESSION 1

Setting up your notebook

Part A of the pre-course self-check · Python, JupyterLab, Qiskit

Six steps to one notebook cell that prints a READY line. 15 to 25 minutes on a normal laptop; installation issues on a company laptop may take longer, which is why this comes first.
Everything installed is public, open-source software.

qubit-lab.ch

qubit-lab.ch
contact@qubit-lab.ch

September 2026 · © 2026 qubit-lab.ch · Proprietary content

Part A — A working notebook

REQUIRED

The goal of this part is one notebook cell that runs and prints a `READY` line. Follow the steps in order. On a normal laptop with a normal connection this takes 15 to 25 minutes, most of it waiting for the download.

> **Company laptop locked down? Use your private laptop.** Everything installed here is public, open-source software from python.org and PyPI: nothing is proprietary and nothing needs an account. The course uses **no corporate data** of any kind, so nothing from your employer ever needs to touch the machine. If your corporate laptop blocks installers or `pip`, do not spend hours with IT; run the course on a private laptop instead. That is the normal case for several participants, and it is the laptop you should bring on the first morning.

Step 1 · Check or install Python

Open a terminal (macOS: Terminal; Windows: Windows Terminal or PowerShell) and type:

```
python3 --version        # macOS / Linux
py --version             # Windows
```

You need Python **3.10 to 3.13**. If you get a version in that range, go to Step 2. Otherwise install Python from python.org/downloads. On Windows, tick **"Add python.exe to PATH"** on the first installer screen, then close and reopen the terminal. On macOS the installer works as is; Homebrew users can run `brew install python@3.13` instead.

Step 2 · Create a project folder and a virtual environment

A virtual environment is a private set of Python packages for this course. It keeps the course installation separate from anything else on your machine, and it is what we use during the sessions.

macOS / Linux:

```
mkdir ~/quantum-kickstart
cd ~/quantum-kickstart
python3 -m venv .venv
source .venv/bin/activate
```

Windows (PowerShell):

```
mkdir $HOME\quantum-kickstart
cd $HOME\quantum-kickstart
py -m venv .venv
.venv\Scripts\activate
```

After the last command the prompt starts with `(.venv)`. That prefix means the environment is active; every `pip` and `jupyter` command from now on refers to it.

> **Windows: use the native installer, not WSL2.** For this course the plain Windows Python from python.org is the simplest path; a Linux subsystem adds nothing the notebooks need and makes JupyterLab harder to reach. If your machine only allows WSL2 with Ubuntu, follow the macOS / Linux commands inside Ubuntu; `jupyter lab` then prints a `http://localhost:8888/lab?token=...` URL that you open in your normal Windows browser.

> **Windows says "running scripts is disabled on this system".** PowerShell blocks the activate script by default. Run `Set-ExecutionPolicy -Scope CurrentUser RemoteSigned` once, answer `Y`, then repeat the activate command.

Step 3 · Install the packages

With the environment active:

```
pip install --upgrade pip
pip install "qiskit>=2,<3" numpy matplotlib jupyterlab ipykernel pylatexenc
```

This downloads roughly 300 MB and takes two to five minutes. The last lines should read `Successfully installed ...` followed by a long list of packages. A warning about a newer pip version is harmless.

Optional, and worth trying now that the environment is active: the noise-model exercises use qiskit-aer, the framework comparison in Session 1 shows the same circuit in PennyLane and Cirq, and the Session 5 algorithms use scipy and scikit-learn.

```
pip install qiskit-aer pennylane cirq scipy scikit-learn
```

If one of these fails to install on your machine, everything else still works; Cirq is the one most likely to be blocked on a company laptop, and the course does not depend on it.

Step 4 · Start JupyterLab and create a notebook

Still in the same terminal:

```
jupyter lab
```

Your browser opens at `http://localhost:8888/lab`. If it does not, copy the URL printed in the terminal (it contains a token) into the browser. Leave the terminal window open; closing it stops the notebook server.

In the launcher page click **Python 3 (ipykernel)** under Notebook. An empty notebook opens with one cell. Rename it via File → Rename Notebook to `setup-check.ipynb`.

Step 5 · Run the test cell

Paste the following into the first cell and press **Shift+Enter**. The same code is in `kickstart-s0-setup-check.ipynb`, which came with this document; opening that file in JupyterLab and running its first cell is equivalent.

```
import sys, platform
import numpy as np
import matplotlib
import qiskit
from qiskit import QuantumCircuit
from qiskit.quantum_info import Statevector
from qiskit.primitives import StatevectorSampler

print("python    ", sys.version.split()[0], "on", platform.system())
print("numpy     ", np.__version__)
print("matplotlib", matplotlib.__version__)
print("qiskit    ", qiskit.__version__)
assert int(qiskit.__version__.split(".")[0]) >= 2, "This course needs Qiskit 2.x"

# 1. One gate, computed twice: by hand with NumPy and by Qiskit
H = np.array([[1, 1], [1, -1]]) / np.sqrt(2)
psi = H @ np.array([1, 0])
qc = QuantumCircuit(1)
qc.h(0)
sv = Statevector(qc).data
assert np.allclose(psi, sv), "NumPy and Qiskit disagree"
print("amplitudes   ", np.round(sv, 4))
print("probabilities", np.round(np.abs(sv) ** 2, 4))

# 2. Measure the circuit 1000 times
qc.measure_all()
print(qc.draw())
job = StatevectorSampler(seed=42).run([qc], shots=1000)
counts = job.result()[0].data.meas.get_counts()
print("counts       ", dict(sorted(counts.items())))

# 3. The line to send us
print(f"\nREADY | python {sys.version.split()[0]} | qiskit {qiskit.__version__}"
      f" | counts {dict(sorted(counts.items()))}")
```

Expected output. Your Python and package versions will differ; everything else should match, including the counts:

```
python     3.13.9 on Darwin
numpy      2.3.5
matplotlib 3.10.7
qiskit     2.5.2
amplitudes    [0.7071+0.j 0.7071+0.j]
probabilities [0.5 0.5]
        ┌───┐ ░ ┌─┐
     q: ┤ H ├─░─┤M├
        └───┘ ░ └╥┘
meas: 1/═══════════╩═
                   0

counts        {'0': 503, '1': 497}

READY | python 3.13.9 | qiskit 2.5.2 | counts {'0': 503, '1': 497}
```

What just happened is Session 1 in miniature: a gate as a 2×2 matrix, applied to a vector with `@`, cross-checked against Qiskit's own simulator, then measured 1000 times. If you can read most of that code already, the course will feel familiar quickly. If not, that is what the course is for.

> **Copy the** `READY` **line** into your reply to the confirmation e-mail. The counts are identical for everyone who uses the same seed, so they double as a check that the sampler works.

Step 6 · Second cell: a plot

Click the empty cell below the output (or press **B** while the first cell is selected to insert one), paste this and press **Shift+Enter**:

```
from qiskit.visualization import plot_histogram
plot_histogram(counts)
```

A bar chart with two bars near 0.5 appears below the cell. If it does, plotting inside the notebook works. If you only see text like `<Figure size ...>` and no picture, add `%matplotlib inline` as the first line of the cell and run it again. Save the notebook with **Ctrl+S** (macOS: **Cmd+S**). Done.

If something did not work

`pip` **cannot connect or times out.** You are behind a corporate proxy or a firewall that blocks `pypi.org` and `files.pythonhosted.org`. IT can configure the proxy for pip or whitelist those two hosts, but in practice the fastest fix is the private laptop (see the box at the start of Part A).

No laptop you can install on at all. Tell us in your reply and we will sort out an alternative with you before the first morning. One option is Google Colab (a notebook in the browser, Google account needed): open **colab.research.google.com**, create a notebook, run `%pip install "qiskit>=2,<3" pylatexenc -q` in the first cell, then paste the test cell from Step 5 into the second.

You prefer VS Code. Install the Python and Jupyter extensions, open the `quantum-kickstart` folder, create a file `setup-check.ipynb`, and pick the `.venv` interpreter when the notebook asks for a kernel. Steps 5 and 6 then work the same way.

Apple Silicon Macs, Intel Macs, Linux, Windows are all fine. Qiskit 2.x publishes pre-built packages for all of them.

Next time

Each time you come back to the course folder, activate the environment and start JupyterLab:

```
cd ~/quantum-kickstart
source .venv/bin/activate      # Windows: .venv\Scripts\activate
jupyter lab
```

Bring the laptop with this working on the first morning; the session notebooks go into the same folder.

Next, and two cheat sheets. With the READY line in hand, continue with the questions of the self-check, on paper or online at qubit-lab.ch/kickstart/self-check. Two cheat sheets cover everything the course uses from linear algebra and complex numbers: qubit-lab.ch/Quantum_Kickstart_Cheat_Sheet_Vectors_Matrices.pdf and qubit-lab.ch/Quantum_Kickstart_Cheat_Sheet_Complex_Numbers.pdf. Print them and keep them next to the laptop on day 1.