

Pre-course setup

These instructions get you ready with a Python installation with the right packages and Jupyter notebooks.

0. Definitions

- **Shell / terminal:** on macOS and Linux, *Terminal*; on Windows, a specific program called *Miniforge Prompt* (installed in step 1) — not `cmd.exe`, not PowerShell.
- **Environment:** an isolated set of Python and its packages (libraries), version-pinned, that doesn't interfere with anything else on your machine. This course uses one environment, named `datacq`, defined by a file called `environment.yml`.

All course materials — this guide, `environment.yml`, `precourse/verify_setup.py`, and later the weekly lab notebooks and data — are distributed as zip bundles on the course's **IM2AG Moodle** page at <https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=721> (log in with your UGA credentials).

1. Install Miniforge (gives you Python + conda)

`conda` is the tool that reads `environment.yml` and installs the exact package versions the course needs. **Miniforge** is a lightweight installer for `conda`: <https://github.com/conda-forge/miniforge>.

macOS

1. In Terminal, download and run the installer (it picks Apple Silicon or Intel automatically):

```
BASE=https://github.com/conda-forge/miniforge/releases/latest/download
curl -fL -o Miniforge3.sh "$BASE/Miniforge3-MacOSX-$(uname -m).sh"
bash Miniforge3.sh
```

2. Accept the license, accept the default install location, and answer **yes** when asked whether to **initialize Miniforge3** — this last one matters, it is what puts `conda` on your `PATH`.
3. Open a new Terminal window (the old one hasn't reloaded your shell config) and check:

```
conda --version
```

“command not found” means the initialization question in step 2 was answered **no** — re-run the installer.

Windows

1. Download the installer: [Miniforge3-Windows-x86_64.exe](#)
2. Run it, choosing “**Just Me**” and the default location. On the “Advanced Options” screen you can leave both checkboxes unchecked — Miniforge gives you its own shortcut instead of modifying your system `PATH`.
3. From here on, **do not use `cmd.exe` or PowerShell for this course**. Open the Start Menu, type **Miniforge Prompt**, and use that program for every command in this guide.

4. In the Miniforge Prompt, check:

```
conda --version
```

Linux

1. Download and run the installer:

```
BASE=https://github.com/conda-forge/miniforge/releases/latest/download
curl -fL -o Miniforge3.sh "$BASE/Miniforge3-Linux-$(uname -m).sh"
bash Miniforge3.sh
```

2. Accept the license, accept the default location, and answer **yes** to initializing Miniforge3.
3. Open a new terminal and check:

```
conda --version
```

(Alternative official download page: <https://conda-forge.org/download/>)

2. Create the course environment

From here on, every command goes in the same shell you used in step 1 (Terminal on macOS/Linux, **Miniforge Prompt** on Windows).

1. Extract the zip you downloaded from Moodle, and `cd` into the extracted folder — the one that directly contains `environment.yml` and a `precourse/` subfolder. For example:

```
cd Downloads/datacq-setup
```

2. Create the environment (5–10 minutes, lots of output):

```
conda env create -f environment.yml
```

3. **Activate** it — you do this once per new shell, every time you work on this course:

```
conda activate datacq
```

Your prompt should now start with `(datacq)`.

3. Launch Jupyter and run the verification script

Jupyter is the notebook app the labs use: cells of code you run interactively in your browser. (If you prefer VS Code with the Python and Jupyter extensions, that works too — just select the `datacq` kernel.)

1. In the same (`datacq`-activated) shell:

```
jupyter notebook
```

This opens a browser tab showing a file listing; if it doesn't open automatically, paste the `http://localhost:8888/...` URL printed in the terminal. Once you see the file listing, Jupyter works — stop it with `Ctrl+C`.

2. Back in the shell, run the verification script:

```
python precourse/verify_setup.py
```

3. **Success** — one **ok:** line per package, ending in:

```
SETUP OK
```

4. If instead you see one or more **FAIL:** lines and **SETUP INCOMPLETE**, read the failing package name(s), and re-check steps 1–2 above.

4. Google Colab alternative

You can do the entire course in **Google Colab**, a free, browser-based Jupyter environment that needs no install: <https://colab.research.google.com>.

1. Sign in with a Google account and create a notebook (**File** → **New notebook**).
2. In the first cell, install the packages Colab lacks:

```
!pip install duckdb mlxtend ucimlrepo
```

3. Copy the contents of `verify_setup.py` into a new cell and run that, rather than running it as a script.

Colab will show a red box below the output saying **An exception has occurred, use %tb to see the full traceback. This is expected:** the script exits with a status code at the end, which notebooks report as an exception. Read the line **above** the box — **SETUP OK** means you're done.

Local setup is the preferred way to work.

5. For week 3

Milestone M0 is a short Moodle form, due **the evening before the week-3 lab**.

- **GitHub account** (mandatory for everyone): create one at <https://github.com>. (We will need it later: GitHub allows only 60 unauthenticated requests per hour, counted per IP address).
- **TMDB API key** (if you plan to choose TMDB as your project's data source): register a free account and request an API key from <https://developer.themoviedb.org/docs/getting-started>.
- **Twitch account with 2FA enabled** (if you plan to do the optional week-3 OAuth2 exercise, or choose Twitch as your project's data source): two-factor authentication is required before you can register a Twitch application.