

# Guide: Setting Up a Conda Environment with Jupyter Notebook and Essential Libraries

Marco Fronzi

## Important Resources for the Course

All codes used in the lectures, labs, and quizzes are available at:

[https://github.com/spostnova/PHYS3034\\_CompPhys](https://github.com/spostnova/PHYS3034_CompPhys)

In this repository, you'll also find the file `environment.yml`, which you can use to automatically replicate the full virtual environment needed for the course.

## 1. Installing Miniconda (Recommended)

Miniconda is a lightweight Python distribution that includes the bare minimum to manage environments and packages with Conda. It works on all platforms.

### Step-by-step installation

#### 1. Visit the download page:

<https://docs.conda.io/en/latest/miniconda.html>

#### 2. Choose the installer for your system:

- **Windows:** Download the `.exe` file for your system (64-bit recommended).
- **macOS:** Choose the Intel or Apple Silicon version depending on your hardware.
- **Linux:** Download the `.sh` script.

#### 3. Run the installer:

- On Windows: Double-click the downloaded file and follow the on-screen instructions.

- On macOS/Linux: Open a terminal, navigate to the folder where the file is located and run:

```
bash Miniconda3-latest-MacOSX-x86_64.sh
```

- Follow the prompts. Accept the license, choose the install location, and allow it to initialize Conda.

## Check the installation

Once done, close and reopen the terminal or Anaconda Prompt and run:

```
conda --version
```

You should see something like `conda 24.1.2` (or similar).

## 2. (Alternative) Installing Anaconda

Anaconda is a full-featured Python distribution that comes preloaded with hundreds of packages. It's much larger in size but simpler for those who want everything included.

- Download from: <https://www.anaconda.com/products/distribution>
- Choose the version for your OS and run the installer.
- After installation, open the `Anaconda Prompt` (Windows) or terminal (macOS/Linux).

## 3. Clone the Course Repository

Clone the GitHub repository that contains all course material and the `environment.yml` file:

```
git clone https://github.com/fronzi/ComputationalPhysicsPython.git  
cd ComputationalPhysicsPython
```

If you don't have Git, you can also download the ZIP file from the GitHub page and extract it.

## 4. Create the Virtual Environment Using the environment.yml File

This will set up all required packages exactly as intended for the course.

### Step-by-step

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

This will create a new environment, typically with a name like `CompPhys`. To activate it, run:

```
conda activate CompPhys
```

You can verify everything was installed with:

```
conda list
```

## 5. Start Jupyter Notebook

While the environment is activated, launch Jupyter Notebook with:

```
jupyter notebook
```

A browser window will open where you can create or open notebooks. Make sure the selected kernel is `CompPhys`.

## 6. Optional: Add the Environment to Jupyter Manually

If for some reason the environment is not appearing in the Jupyter kernel list, you can add it manually:

```
python -m ipykernel install --user --name=CompPhys
```

Now youll see `CompPhys` as an option when selecting a kernel in your notebooks.

## 7. Recreate Environment Anywhere

If a student wants to replicate the setup on another machine, they only need:

- The `environment.yml` file
- To run:

```
conda env create -f environment.yml
conda activate CompPhys
jupyter notebook
```

## Troubleshooting Tips

- If you get an error like “conda not found”, try closing and reopening your terminal.
- On macOS/Linux, you may need to run: `source / .bashrc` or `source / .zshrc`
- Always activate your environment before running any notebooks.
- Use `conda info --envs` to list all environments on your system.