Setting Up Julia, JupyterLab, and Pluto on Linux

Department of Physics, University of Southern Maine

September 2026

These instructions walk you through installing Julia on a Linux computer (Ubuntu, Pop!_OS, Fedora, and similar) and setting it up so that you can open JupyterLab or Pluto notebooks with a single command in the terminal. Wherever you see a `$` prompt, type the command into your terminal; wherever you see `julia>` or `pkg>`, type it into the Julia REPL.

1 Install Julia

Do not use your distribution's package manager (`apt`, `dnf`, etc.) to install Julia; those packages are usually several versions out of date. Instead, open a terminal and run the official installer:

```
$ curl -fsSL https://install.julialang.org | sh
```

This installs `juliaup`, the official Julia version manager, along with the current release of Julia, and adds it to your PATH. Accept the defaults when prompted. Close and reopen the terminal (or run `source ~/.bashrc`), then confirm the installation:

```
$ julia --version
julia version 1.13.0
```

If you already have Julia installed through `juliaup` and want to update, run `juliaup update`.

2 Add the packages

Start Julia by typing `julia` in the terminal. At the `julia>` prompt, press `]` to enter package mode and add the two packages you need:

```
pkg> add IJulia Pluto
```

`IJulia` lets Jupyter run Julia code; `Pluto` is a self-contained notebook system that needs nothing else. Press Backspace to return to the `julia>` prompt, then load `IJulia` once so that it registers the "Julia 1.13" kernel with Jupyter:

```
julia> using IJulia
```

Packages you install this way go into the default environment, located at `~/.julia/environments/v1.13`. Any other packages you need for coursework (for example `CairoMakie`, `DifferentialEquations`, or `LsqFit`) can be added the same way.

3 Install JupyterLab

The easiest option is to let `IJulia` install and manage its own copy of JupyterLab, so there is no separate Python installation to keep in sync. Still in the Julia REPL, run:

```
julia> using IJulia
julia> jupyterlab()
```

The first time you run this, Julia will ask whether it may install JupyterLab through Conda; answer `y`. The download takes a few minutes. When it finishes, JupyterLab opens in your web browser. To stop it, close the browser tab and press Ctrl-C in the terminal, then type `exit()` to leave Julia.

4 One-word launchers for the terminal

To avoid typing the commands above every time, define two aliases in your shell's startup file. Most Linux distributions use Bash, whose startup file is `~/.bashrc`; if you use Zsh, edit `~/.zshrc` instead. Open the file in a text editor, for example:

```
$ nano ~/.bashrc
```

Add the following two lines at the end, then save (Ctrl-O, Enter) and exit (Ctrl-X):

```
alias jlab='julia -e "using IJulia; jupyterlab(dir=pwd(), detached=true)"'
alias pluto='julia -e "using Pluto; Pluto.run()"'
```

Close and reopen the terminal (or run `source ~/.bashrc`). From now on:

- `jlab` opens JupyterLab in your browser, rooted in whatever folder you are currently in, and gives you your prompt back right away. (Remove `detached=true` if you would rather keep it attached so that Ctrl-C shuts it down.)
- `pluto` starts Pluto and opens it in your browser. Press Ctrl-C in the terminal to stop it.

A typical workflow is to `cd` into the folder for a homework assignment and then type `jlab` or `pluto`.

5 If you already have Jupyter installed

If you have JupyterLab installed through your distribution, `pip`, or Anaconda, you can keep using it: just run `jupyter lab` as usual. `IJulia` writes its kernel definition to the standard location, `~/.local/share/jupyter/kernels/julia` so your existing Jupyter will list "Julia 1.13" in its kernel menu. In that case you can skip Section 3 and change the `jlab` alias to simply `alias jlab='jupyter lab'`.

6 Troubleshooting

- **`julia: command not found`.** Close and reopen the terminal after installing, or run `source ~/.bashrc`; the installer adds Julia to your PATH but the current shell does not see the change.
- **The Julia kernel does not appear in Jupyter.** In the Julia REPL, run `using IJulia; IJulia.installkernel("J` to re-register it.
- **The browser does not open automatically.** Julia uses `xdg-open` to launch your default browser. If nothing appears, copy the `http://localhost:...` address printed in the terminal and paste it into your browser. On a minimal system you may need `sudo apt install xdg-utils` (or the equivalent for your distribution).
- **`jlab` or `pluto` is not recognized.** Check that the aliases were saved to the startup file for the shell you actually use (`echo $SHELL` tells you which one), and reopen the terminal.