Setting Up JupyterLab and Pluto with Julia 1.13

Physics 321

14 September 2026

1 Packages to add

Only two packages are needed in the default environment. In the Julia REPL, press `]` to enter package mode and run:

```
pkg> add IJulia Pluto
```

`IJulia` provides the Jupyter kernel for Julia; `Pluto` is self-contained and needs nothing else.

After upgrading, the Julia 1.13 default environment (`~/.julia/environments/v1.13`) starts empty, so any packages previously installed under 1.12 (Makie, DifferentialEquations, LsqFit, etc.) must be re-added there as well.

`IJulia` registers a kernelspec when it is installed. To be sure the "Julia 1.13" kernel is registered, load it once in the REPL:

```
julia> using IJulia
```

2 Launching JupyterLab from the terminal

The simplest, most reliable approach is to let `IJulia` manage its own JupyterLab installation through `Conda.jl`, rather than mixing it with a Python-side Jupyter that must be kept in sync. Run this once in the REPL:

```
julia> using IJulia
julia> jupyterlab()   # first run installs JupyterLab into Julia's private Conda env
```

Then add an alias to `~/.zshrc`:

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

Typing `jlab` in any directory opens JupyterLab rooted there and returns the terminal prompt immediately. Omit `detached=true` to keep JupyterLab attached to the terminal so that Ctrl-C stops it.

Alternative: an existing Jupyter installation

If JupyterLab is already installed via Homebrew or a Python virtual environment, simply run `jupyter lab` as usual. The `IJulia` kernelspec is written to the standard Jupyter kernels directory (`~/Library/Jupyter/kernels/julia-1.13` on macOS), so any Jupyter on the machine will find it.

3 Launching Pluto from the terminal

A matching alias for Pluto:

```
alias pluto='julia -e "using Pluto; Pluto.run()"'
```

4 A note on environments

Keep `IJulia` and `Pluto` in the default environment and do actual project work in dedicated project environments. Because Julia stacks environments, a notebook kernel will still see the default environment's packages, so there is no need to reinstall the kernel for each project.