

Frequently Asked Questions

Setting up and running Julia



My Julia code does not run in Zed. What can I do?

The commands in **bold** are in Zed's command palette (`Ctrl+Shift+P`, macOS `Cmd+Shift+P`).

- **Nothing happens when I press `Ctrl+Shift+Enter`.** Save the file with the ending `.jl` and press the shortcut a second time: right after Zed starts, the first press can get lost while Zed looks for its kernels. Each cell has to start with `# %%` at the very beginning of a line.
- **Zed does not list the kernel `optimization-with-julia-1.13`.** Run **repl: refresh kernelspecs**. If it is still missing, repeat steps 3 to 5 of the [first tutorial](#) and look for error messages after `installkernel`.
- **ArgumentError: Package JuMP not found in current path** (or any other package). Zed started another kernel, for example `julia-1.13`, which does not use the workshop environment. Check the kernel selector at the top right of the editor and choose `optimization-with-julia-1.13`. If the kernel is right, the package is not installed yet: run `using Pkg; Pkg.add("JuMP")` in a cell.
- **UndefVarError: ... not defined.** The cell uses a variable from a cell that has not run yet. Run the cells above it first. After **repl: restart**, all variables are gone and you start again from the top.
- **Plots do not appear.** Your kernel was installed without the `zed_plots` part of step 5 in the [first tutorial](#): close Zed and run steps 3 and 5 again. Interactive plots made with `plotly()` cannot be shown in Zed; leave that line out.
- **A model keeps solving and does not stop.** **repl: interrupt** cannot stop a running solver. Use **repl: restart** instead and run your cells again from the top.
- **The kernel does not start after a Julia update.** The kernel still points to the old Julia version, which the update removed. Close Zed and run steps 3 and 5 of the [first tutorial](#) again.
- **The first cell takes minutes.** Julia compiles each package once on your computer, so the first `using JuMP` or `using Plots` is slow, on Windows even slower because Microsoft Defender scans the new files. Let it finish: afterwards, the same commands are fast.
- **An exclamation mark appears in Zed's title bar, and code completion does not work.** Zed opened your folder in Restricted Mode. Click the exclamation mark and trust your workshop folder (only that folder, not your whole user folder).
- **`Ctrl+C` does not copy in Zed's terminal (Windows, Linux).** In the terminal, `Ctrl+C` stops the running program; copy with `Ctrl+Shift+C`.

If Zed itself does not run on your computer, use the VS Code fallback described in the [first tutorial](#). The Jupyter notebooks of this workshop run there as well.

Julia or Zed does not work on my Windows laptop. What can I do?

- **The terminal does not find `julia`.** Open a new terminal; if you use the terminal inside Zed, restart Zed. If that does not help, open *Settings* → *Apps* → *Advanced app settings* → *App execution aliases* and switch on `julia.exe` and `juliaup.exe`. If the Microsoft Store is blocked on your laptop, use the [MSI installer](#) instead.
- **“An Application Control policy has blocked this file”, “Smart App Control has blocked part of this app” or error `0xc0e90002`.** Windows Smart App Control blocks the package files that Julia compiles on your laptop. Switch it off under *Windows Security* → *App & browser control* → *Smart App Control* and keep it off while you work with Julia: switched on again, it can block every package Julia compiles later. Then delete the folder `%USERPROFILE%\julia\compiled\v1.13` and run `using Pkg; Pkg.precompile()` in the workshop environment. On a laptop managed by your university or employer, only the IT department can allow these files; until then, start Julia with `julia --pkgimages=no --project=optimization-workshop` in the [first tutorial](#) and add `--pkgimages=no` right after the `--project` argument of its `installkernel` call. Packages then load more slowly, but they load.
- **Precompiling IJulia fails with an `EPIPE` error.** Run `using IJulia` again: the error is known on Windows and usually disappears on the second attempt.
- **Your laptop has a Snapdragon (ARM) processor.** Julia runs there in x64 emulation: the workshop setup should work the same way, only slower.
- **Your workshop folder lies in OneDrive.** Synced folders can block Julia from replacing files. Use a plain folder such as `C:\Users\<name>\julia-workshop`.

Still stuck? Ask the course chatbot on this website: it knows these steps.