Installing Julia

Binary file

Download the Julia binary from the official website. The binary is available for Windows, macOS, and Linux.

- **Recommended version** : *Current stable release: v1.10.4 (June 4, 2024)*
- Install a **64-bits** binary
- Be aware of installing v1.11, it is still a release candidate version and not all packages may work out of the box.

Via CLI

- Windows:

```
> winget install julia -s msstore
```

- Linux & MacOS:

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

Juliaup - version manager (RECOMMENDED)

Juliaup is a version manager that allows you to easily install and switch to different Julia versions.

Juliaup installation

- Windows:

```
> winget install julia -s msstore
```

- Mac & Linux:

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

Julia installation

You can install any version of Julia with the Juliaup CLI tools. For example, to install julia-1.10.4 open your terminal and type

```
> juliaup add 1.10.4
```

If you have several Julia versions installed, you can switch versions by openning julia as e.g.

```
> julia +1.9.1
```

or make it the default version as

```
> juliaup default 1.9.1
```

Check `juliaup` GitHub page for more config options.

Using Julia

The recommended way of developing code in Julia is using the VSCode IDE.

Julia with VSCode

Installing Julia plug-in in VSCode

To run Julia interactively in VSC you will need to install Julia's plugin by following the steps in the snapshot:

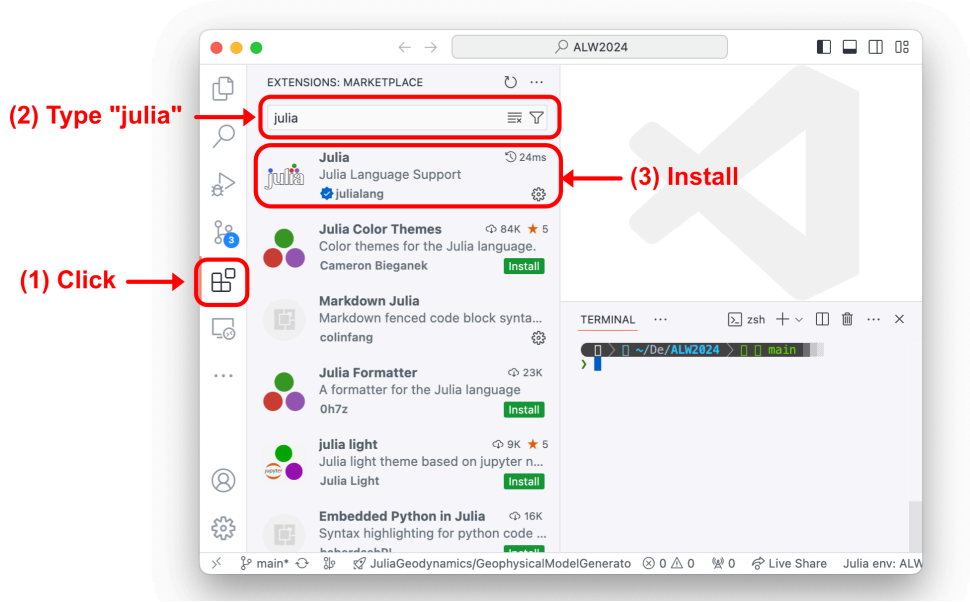

Figure 1: How to install Julia VSCode plugin

Other usefull plugins:

- GitHub Pull Request
- GitLense
- Remote Explorer
- Live Share
- LaTeX / Typst
- Markdown
- Jupyter
- Quarto
- etc...

Starting an interactive Julia session in VSCode

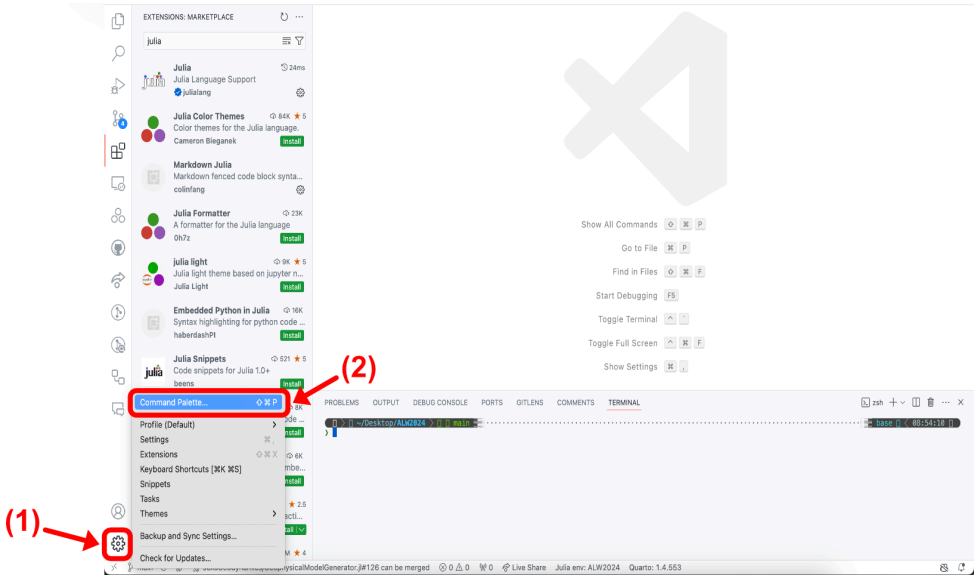

Figure 2: How to open an interactive Julia session in VSCode

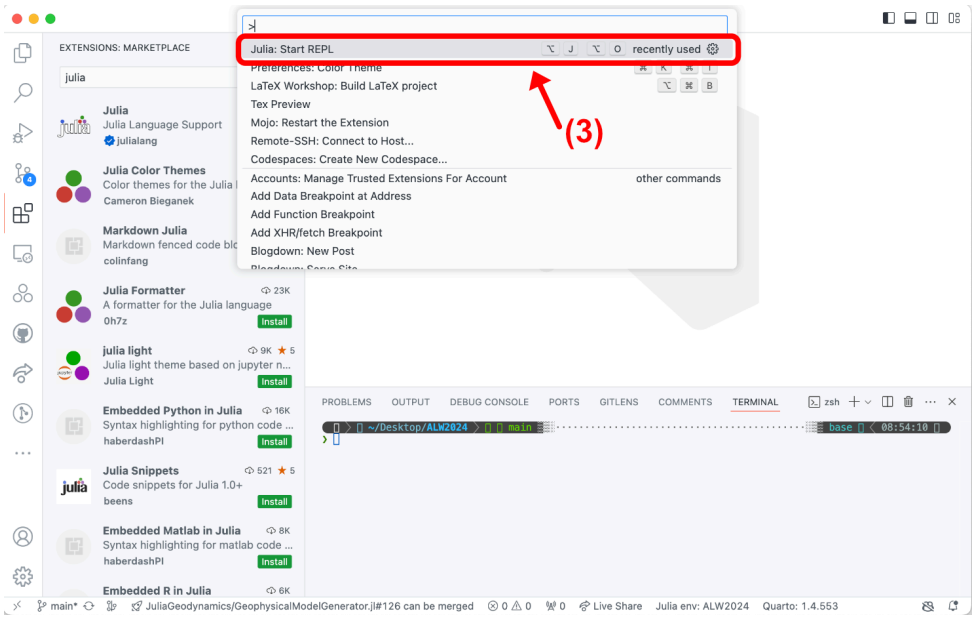

Figure 3: How to open an interactive Julia session in VSCode

Julia from the terminal

Interactive session

1. Add `julia` binary to your environment path
2. Open terminal
3. Type `julia` to open the REPL

Launching a script

1. Add `julia` binary to your environment path
2. Open terminal
3. Type `julia myScript.jl`. More info here.

Installing registered packages

Packages that are registered in Julia's general registry can easily be installed with Julia's built-in package manager. To enter the package manager mode, simply open an interactive julia session and type `]` in the REPL. Once inside the package manager, a given package can be installed by typing `add MyPkg`. For example, to install `GLMakie.jl`, one of the plotting packages, just type

```
] add GLMakie
```

Specific versions can also be insalled, for example, to install v0.5 of GLMakie

```
] add GLMakie@0.5
```

and also specific branches from the host repository

```
] add GLMakie#myBranch
```

Packages can also be installed outside the package manager in a more verbose way

```
import Pkg; Pkg.add("GLMakie")
```

Required packages

- GLMakie.jl/CairoMakie (or Plots.jl) for visualization.
- IJulia to run Jupyter notebooks with the Julia kernel
- ParallelStencil.jl (Multithreading, CUDA, AMDGPU) or KernelAbstractions.jl (Multithreading, CUDA, AMDGPU, M-series chips) for parallel programming.
- ImplicitGlobalGrid.jl MPI communication

Optional

- GeophysicalModelGenerator.jl
- GMT.jl
- Interpolations.jl

If there is time...

- JustRelax.jl
- GeoParams.jl
- JustPIC.jl