

# Julia Install Guide

This is not really an installation guide. It's a set of steps and links to helpful information on those steps.

## Background

These days Julia comes as a stand-alone package which has a build-in installation of Python, which is used for some tasks. You can use your own version of Python, but that complicates life, so best to start with the default.

Julia, by itself, is fully functional, but most people find it easier to interact with it through an IDE (like Rstudio). the one of choice these days is VS Code.

Finally, to get full utility with Julia, you'll need to install some extra packages. This is really easy, but it's helpful to have a list of go-to packages to start with.

This guide is aimed at MacOS. I often use it on Linux, but the instructions for that are pretty standard, except the default Julia installation is local to a user, not global.

## Step 1: Install Julia

Current recommended version as of writing this is 1.5.3. Version 1.6 is soon, but I don't recommend that newcomers hit the bleeding edge.

Download from <https://julialang.org/downloads/>

Platform install instructions are at <https://julialang.org/downloads/platform/>

On a Mac, you can test it by opening the application (from your application folder) or running the Julia command in a shell (command window) by putting it on your path using

```
ln -s /Applications/Julia-1.5.app/Contents/Resources/julia/bin/julia/usr/local/bin/julia
```

## Step 2: Install VS Code

VS Code is a separate application that is designed as an Integrated Development Environment (IDE) for almost any programming language. It has specific extensions that customise it to particular languages like Julia.

Instructions on installation and configuration to use Julia are here:

<https://www.julia-vscode.org/docs/dev/gettingstarted/#Installation-and-Configuration-1>

I also thought this web page was useful:

<https://techtok.com/julia-vscode/>

Both of those pages give examples of running Julia in VS Code. Have a go!

There is more documentation for Julia in VS Code here:

<https://www.julia-vscode.org/>

There are also alternative ways to interact with Julia through workbook environments like Jupyter, e.g., see

<https://datatofish.com/add-julia-to-jupyter/>

Note, however, that I primarily use Julia directly in a command window, so there is no need to use VS Code if you don't care for it.

## Step 3: Packages

We mainly install packages through Julia itself. You can do this directly, for example:

```
using Pkg
Pkg.status()
```

However there is a shorthand. Hit the `]`  key, and it will open a package manager mode.

```
]
(@v1.5) pkg> status
```

To add a package, got to the package manager mode and just type

```
add PACKAGENAME
```

I have a quite long list of packages installed, but I would recommend a smaller list to start. In my view essential packages are

- Cairo (bindings to the Cairo backend for plotting)
- Blink (useful for displaying some information)
- PyCall (to call Python code)
- Plots and PyPlot (the first is the main plotting package, and the second a Matlab/Python-like package)
- ORCA (helpful for saving plots to files)
- DataFrames and DelimitedFiles and CSV (for working with data)
- Statistics and Distributions and StatsBase (guess)
- Printf (a printf library for C lovers)
- PackageCompiler (we'll use this later)
- Polylogarithms (I just want people to play with my package :)

BTW, Julia packages are hosted on Github, so they are easy to play with directly, and when you install it, the code will all be placed in your local julia directory, so you can easily inspect it.

When you install a package Julia will automatically install all dependencies (most of the time).

The first time you use many packages, e.g, PyCall, it will automatically "build" it. You might need to rebuild in some circumstances, but that is hopefully rare.

## Step 4: Fun

Let's do a plot. Type in the command window:

```
using Plots
pyplot() # to select PyPlot style plots
plot([1,2,3], [4,5,6])
```

<https://docs.juliaplots.org/latest/install/>

