

# How to install Julia?

# How to install Julia?

- Download a proper file from the [website](#) :

Julia version (v1.11.0 October 7, 2024) from the [Microsoft Store](#) by running this in the command

```
julia -s msstore
```

on Windows. For Linux and MacOS instructions [click here](#)

Julia will be available via the command line interface.

[Juliaup](#) installation manager, which will automatically install Julia and help keep it up to date. It is also installed. To install different Julia versions see `juliaup --help`.

# How to install Julia?

- We will use version **1.11.0**

Julia version (**v1.11.0** October 7, 2024) from the [Microsoft Store](#) by running this in the command

```
julia -s msstore
```

on Windows. For Linux and MacOS instructions [click here](#)

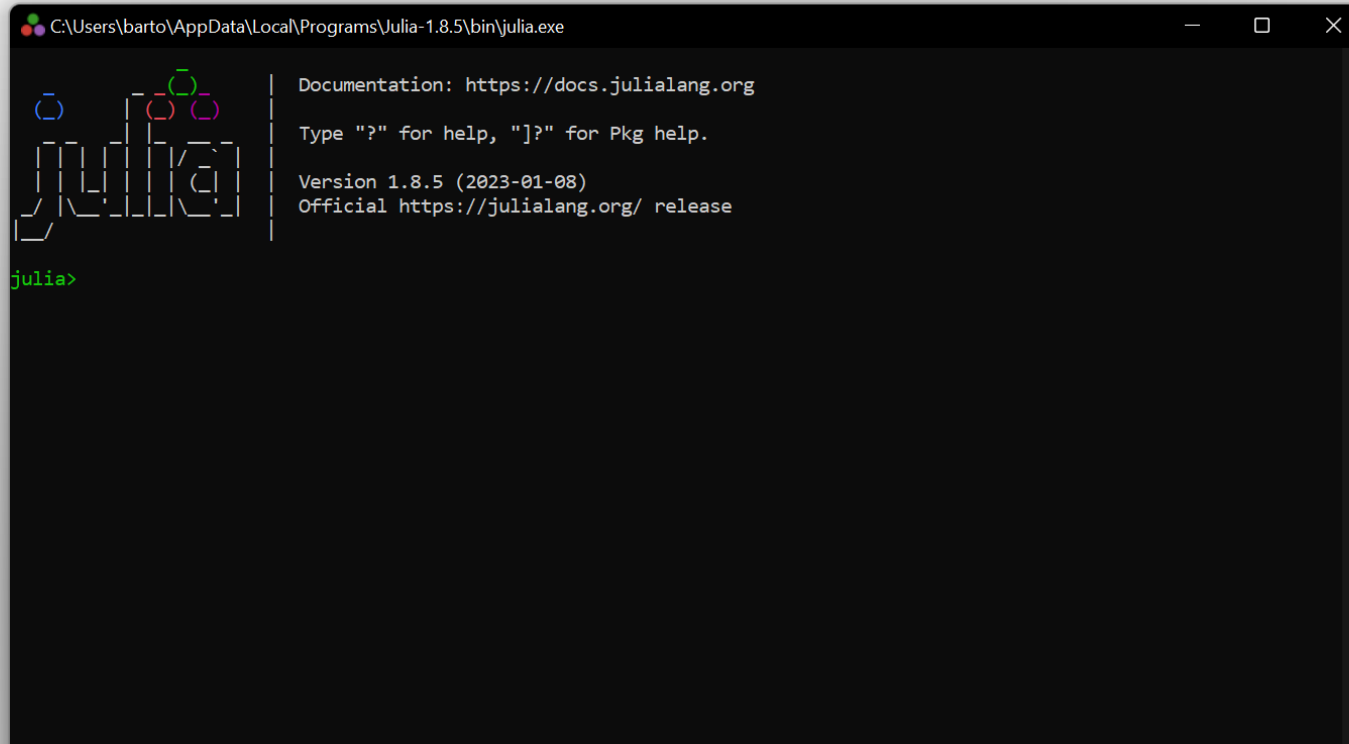
Julia will be available via the command line interface.

[Juliaup](#) installation manager, which will automatically install Julia and help keep it up to date. [Juliaup](#) is also installed. To install different Julia versions see `juliaup --help`.

# How to configure Julia?

- To configure the environment run **REPL**:

Please star us [on GitHub](#). If you use Julia in your research, please [cite us](#). If possible, do consider [sponsoring us](#).

A screenshot of the Julia REPL (Read-Eval-Print Loop) window. The window title is "C:\Users\barto\AppData\Local\Programs\Julia-1.8.5\bin\julia.exe". The window has a dark background. On the left side, there is a logo consisting of the word "julia" in a stylized, blocky font, with the letters colored in blue, green, and red. To the right of the logo, the following text is displayed: "Documentation: https://docs.julialang.org", "Type '?' for help, ']' for Pkg help.", "Version 1.8.5 (2023-01-08)", and "Official https://julialang.org/ release". At the bottom left, the prompt "julia>" is visible in green text.

```
C:\Users\barto\AppData\Local\Programs\Julia-1.8.5\bin\julia.exe

julia> Documentation: https://docs.julialang.org
Type "?" for help, "]" for Pkg help.
Version 1.8.5 (2023-01-08)
Official https://julialang.org/ release

julia>
```

portable)

GitHub

Julia versions, so



# How to configure Julia?

- Firstly, we will install [the necessary packages](#).
- It can be done in two ways:
  - By pressing a “**]**” button in REPL. You will enter Julia’s built-in package manager (note the blue prompt marker):

A screenshot of the Julia REPL (Read-Eval-Print Loop) interface. The window title is "Julia 1.8.5". On the left, there is a stylized logo made of dashed lines and colored brackets. The main area displays the following text: "Documentation: https://docs.julialang.org", "Type '?' for help, ']?' for Pkg help.", "Version 1.8.5 (2023-01-08)", and "Official https://julialang.org/ release". At the bottom, the prompt "(@v1.8) pkg>" is shown in blue, indicating that the package manager has been entered.

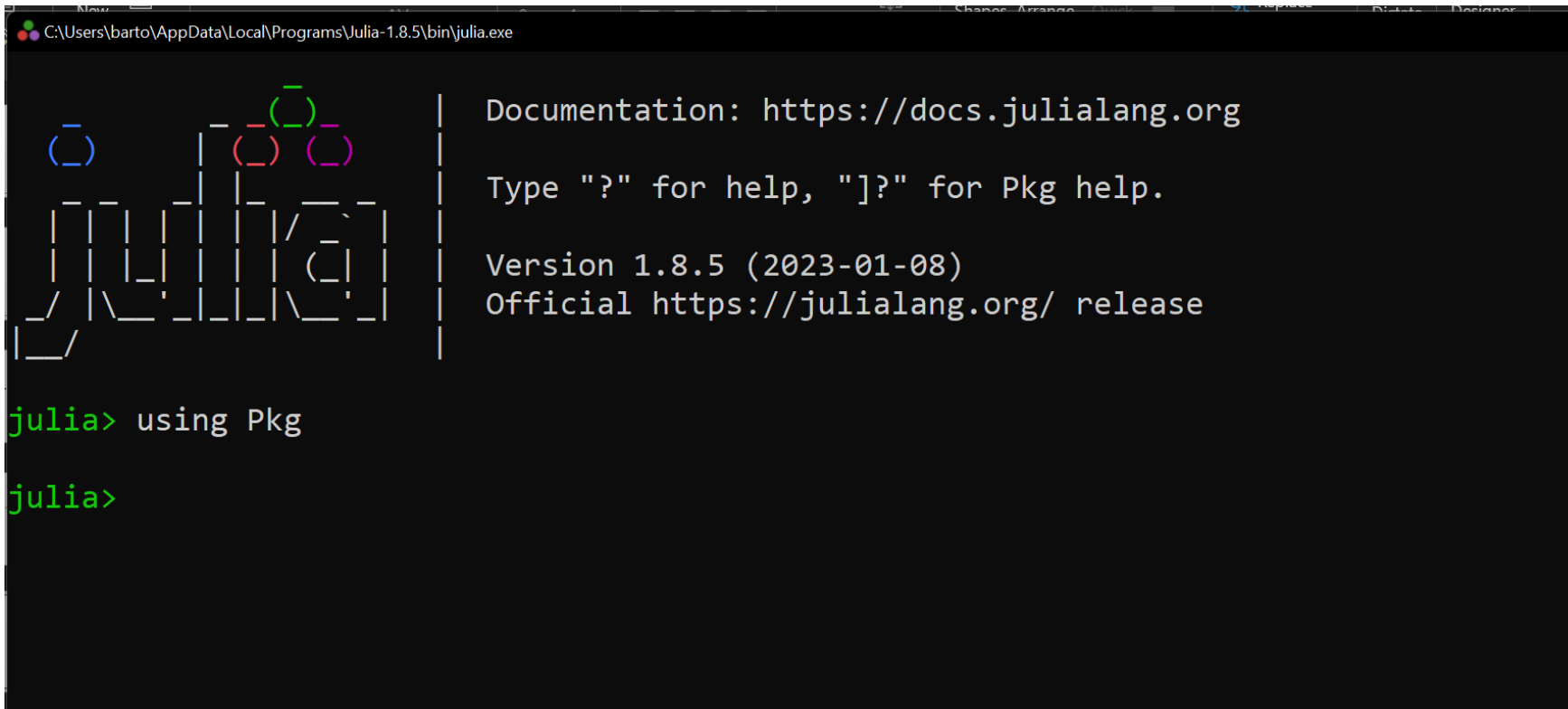
```
Julia 1.8.5

Documentation: https://docs.julialang.org
Type "?" for help, "]"? for Pkg help.
Version 1.8.5 (2023-01-08)
Official https://julialang.org/ release

(@v1.8) pkg>
```

# How to configure Julia?

- Firstly, we will install [the necessary packages](#).
- It can be done in two ways:
  - or by importing the “**Pkg**” package:



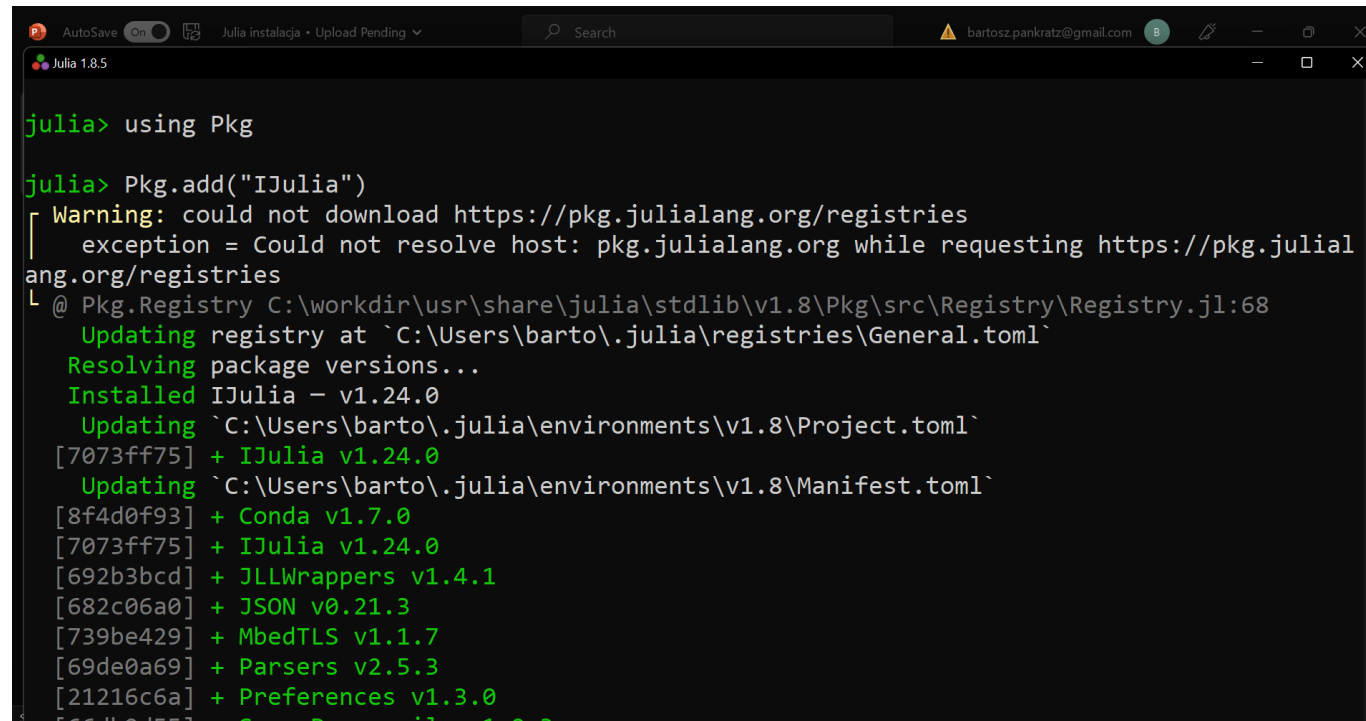
The screenshot shows the Julia REPL interface. On the left, there is a stylized ASCII art logo of the letter 'J' composed of various colored parentheses. To the right of the logo, the following text is displayed:

```
Documentation: https://docs.julialang.org
Type "?" for help, "]"? for Pkg help.
Version 1.8.5 (2023-01-08)
Official https://julialang.org/ release
```

Below this information, the user has entered the command `using Pkg` at the `julia>` prompt. The prompt is now `julia>` again, indicating the command has been executed.

# How to configure Julia?

- We will start with "IJulia" package.
- IJulia is necessary to run Julia's code in Jupyter Notebook format. Note that you need a properly configured Jupyter environment (e.g. installed Anaconda). Let's install IJulia:



```
julia> using Pkg

julia> Pkg.add("IJulia")
[ Warning: could not download https://pkg.julialang.org/registries
  exception = Could not resolve host: pkg.julialang.org while requesting https://pkg.julialang.org/registries
]
@ Pkg.Registry C:\workdir\usr\share\julia\stdlib\v1.8\Pkg\src\Registry\Registry.jl:68
Updating registry at `C:\Users\barto\.julia\registries\General.toml`
Resolving package versions...
Installed IJulia - v1.24.0
Updating `C:\Users\barto\.julia\environments\v1.8\Project.toml`
[7073ff75] + IJulia v1.24.0
Updating `C:\Users\barto\.julia\environments\v1.8\Manifest.toml`
[8f4d0f93] + Conda v1.7.0
[7073ff75] + IJulia v1.24.0
[692b3bcd] + JLLWrappers v1.4.1
[682c06a0] + JSON v0.21.3
[739be429] + MbedTLS v1.1.7
[69de0a69] + Parsers v2.5.3
[21216c6a] + Preferences v1.3.0
[56db0d55] + ProgressMeter v1.0.3
```

# How to configure Julia?

- **"PyCall"** package provides the ability to call Python from the Julia language directly:

```
julia> Pkg.add("PyCall")
  Resolving package versions...
  Installed PyCall - v1.95.1
  Updating `C:\Users\barto\.julia\environments\v1.8\Project.toml`
[438e738f] + PyCall v1.95.1
  Updating `C:\Users\barto\.julia\environments\v1.8\Manifest.toml`
[1914dd2f] + MacroTools v0.5.10
[438e738f] + PyCall v1.95.1
[37e2e46d] + LinearAlgebra
[e66e0078] + CompilerSupportLibraries_jll v1.0.1+0
[4536629a] + OpenBLAS_jll v0.3.20+0
[8e850b90] + libblastrampoline_jll v5.1.1+0
  Building PyCall → `C:\Users\barto\.julia\scratchspaces\44cfe95a-1eb2-52ea-b672-e2afdf69b78f\62f417f6ad727987c755549e9cd88c46578da562\build.log`
  Precompiling project...
  5 dependencies successfully precompiled in 10 seconds. 16 already precompiled.

julia> _
```



# How to configure Julia?

- It is often necessary to set a different version of Python than the default. You can change the Python version by setting the `PYTHON` environment variable to the path of your preferred Python version and then re-running `Pkg.build("PyCall")`:

```
ENV["PYTHON"] = "... path of the python executable ..."  
# ENV["PYTHON"] = raw"C:\Python310-x64\python.exe" # example for Windows, "raw" to not have to escape: "C:\\P  
  
# ENV["PYTHON"] = "/usr/bin/python3.10" # example for *nix  
Pkg.build("PyCall")
```

# How to configure Julia?

- Path to default Anaconda installation on Windows 11:

```
julia> ENV["PYTHON"] = "C:\\Users\\barto\\Anaconda3\\python.exe"
"C:\\Users\\barto\\Anaconda3\\python.exe"

julia> Pkg.build("PyCall")
  Building Conda → `C:\\Users\\barto\\.julia\\scratchspaces\\44cfe95a-1eb2-52ea-b672-e2afdf69b78f\\6e47d11ea2776bc5627421d59cdcc1296c058071\\build.log`
  Building PyCall → `C:\\Users\\barto\\.julia\\scratchspaces\\44cfe95a-1eb2-52ea-b672-e2afdf69b78f\\62f417f6ad727987c755549e9cd88c46578da562\\build.log`
Precompiling project...
 1 dependency successfully precompiled in 8 seconds. 20 already precompiled.
```

# How to configure Julia?

- You can also specify a package version:

```
julia> Pkg.add(name = "DataFrames", version = "1.3.6")  
  Resolving package versions...  
No Changes to `C:\Users\barto\.julia\environments\v1.8\Project.toml`  
No Changes to `C:\Users\barto\.julia\environments\v1.8\Manifest.toml`
```

# How to configure Julia?

- You can check the currently installed packages and their versions with “**Pkg.status()**” command:

```
julia> Pkg.status()
Status `C:\Users\barto\.julia\environments\v1.8\Project.toml`
 [b7f77d8d] ArcadeLearningEnvironment v0.2.4
 [fbb218c0] BSON v0.3.6
 [a93c6f00] DataFrames v1.3.6
 [587475ba] Flux v0.13.4
 [e15a9946] GridWorlds v0.5.0
 [7073ff75] IJulia v1.24.0
 [a09fc81d] ImageCore v0.9.4
 [916415d5] Images v0.25.2
 [91a5bcd] Plots v1.38.2
 [438e738f] PyCall v1.95.1
 [d330b81b] PyPlot v2.11.0
 [158674fc] ReinforcementLearning v0.10.1
 [e575027e] ReinforcementLearningBase v0.9.7
 [25e41dd2] ReinforcementLearningEnvironments v0.6.12
Info Packages marked with [?] and [?] have new versions available, but those with [?] are restricted
by compatibility constraints from upgrading. To see why use `status --outdated`
```

# How to configure Julia?

- To run notebooks correctly, the following packages and versions are required:

- **BSON** v0.3.9
- **CUDA** v5.5.2
- **DSP** v0.7.10
- **DataFrames** v1.7.0
- **Flux** v0.14.22
- **IJulia** v1.25.0
- **ImageCore** v0.10.2
- **Images** v0.26.1
- **JLD2** v0.4.53
- **LaTeXStrings** v1.3.1
- **MLDatasets** v0.7.18
- **Metalhead** v0.9.4
- **Optimisers** v0.3.3
- **ParameterSchedulers** v0.4.2
- **Plots** v1.40.7
- **PyCall** v1.96.4
- **StatsBase** v0.34.2
- **TestImages** v1.8.0
- **WAV** v1.2.0
- **Zygote** v0.6.71