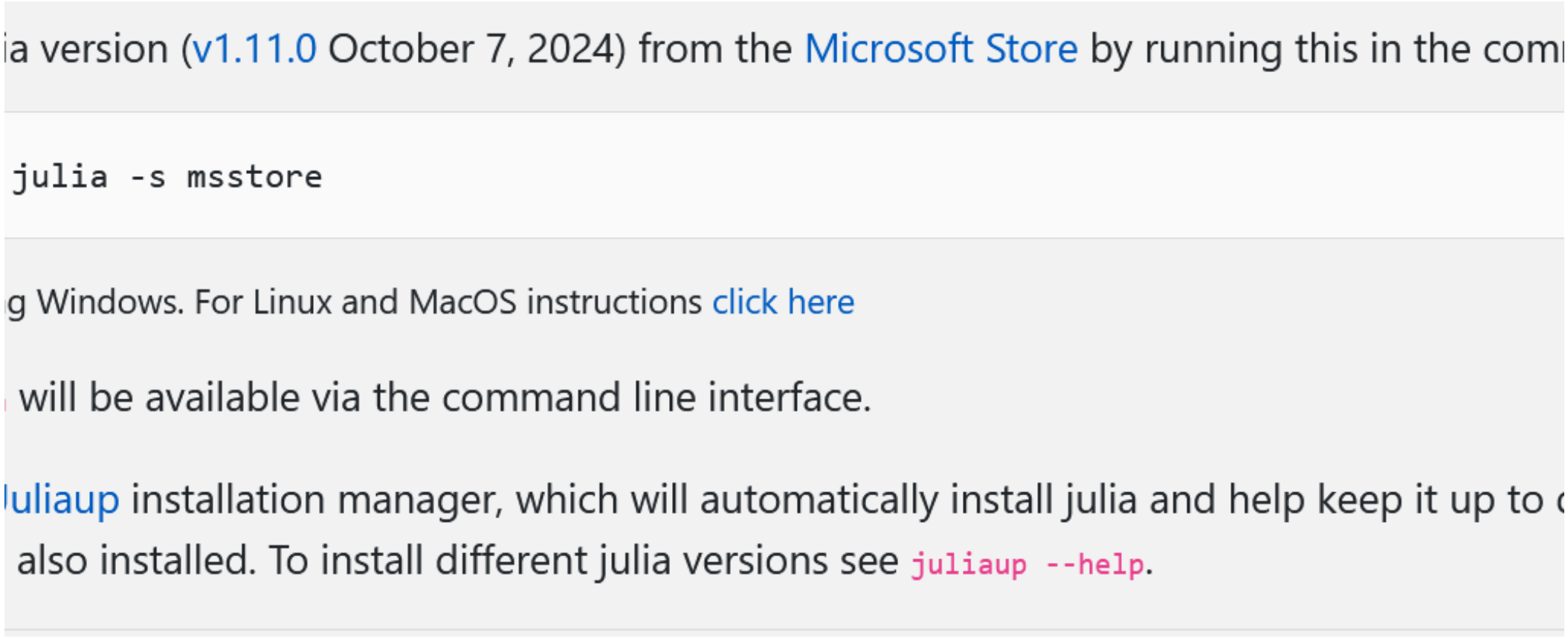
How to install Julia?

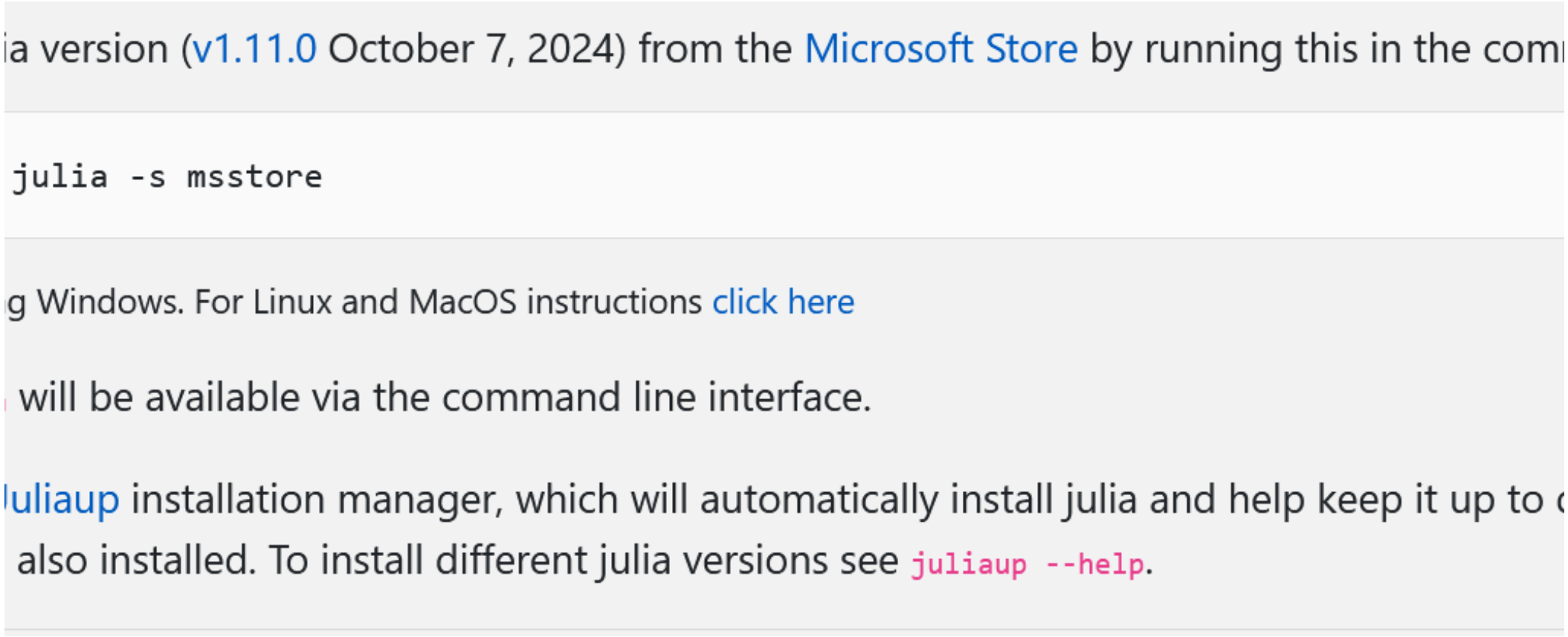
How to install Julia?

- Download a proper file from the website :



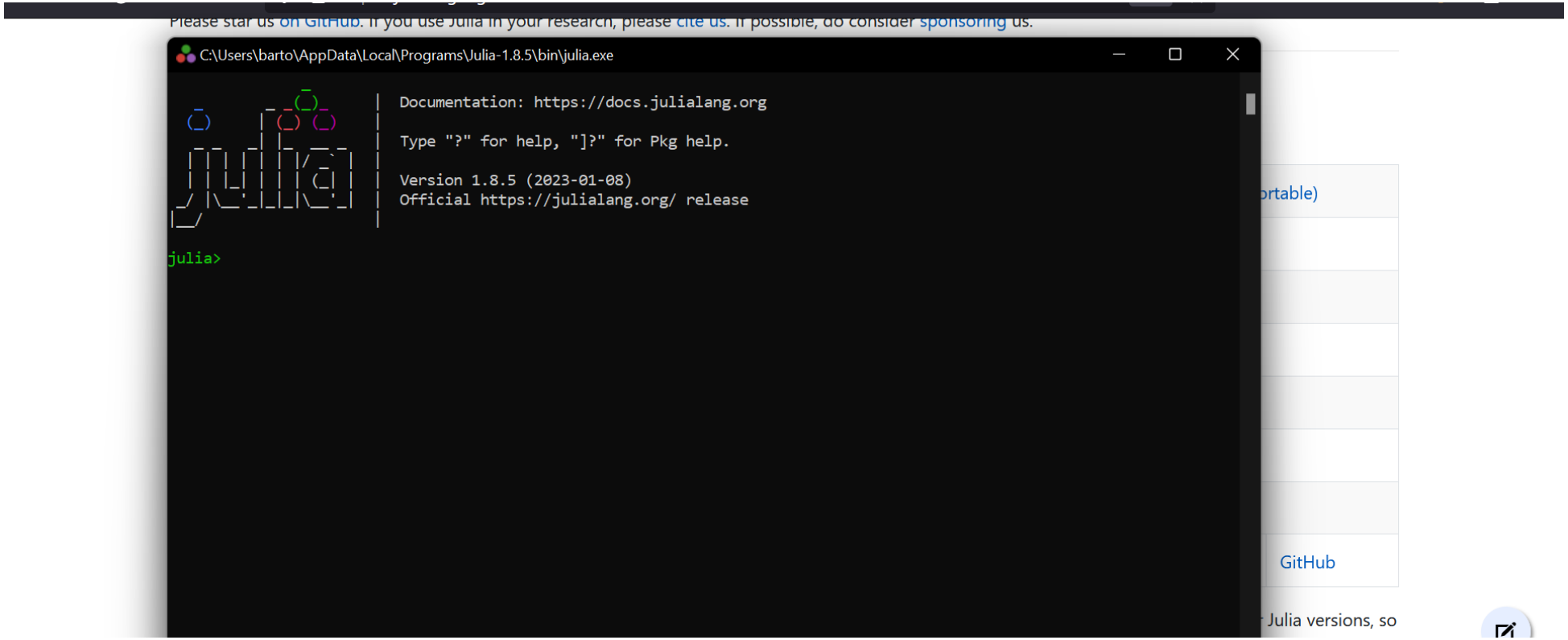
How to install Julia?

- We will use version **1.11.0**



How to configure Julia?

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

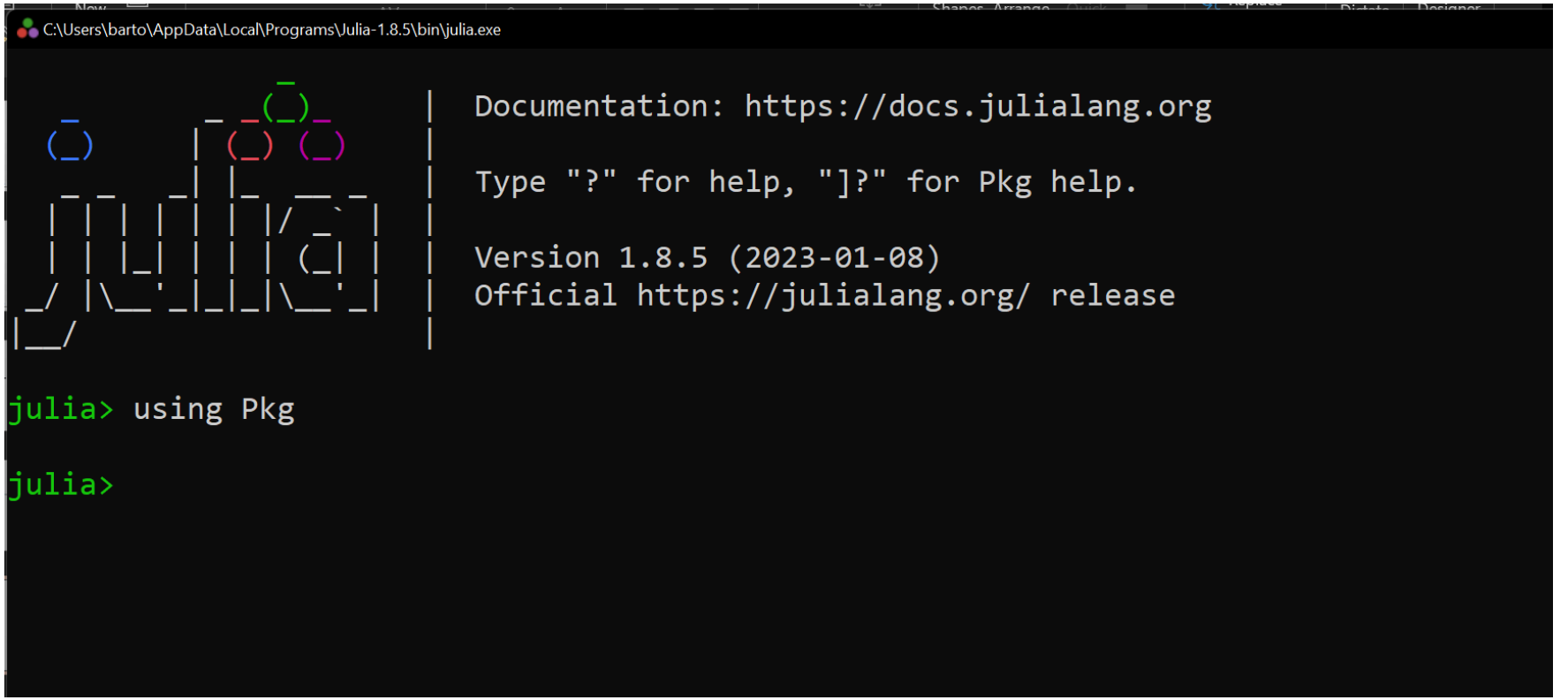


How to configure Julia?

- 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):

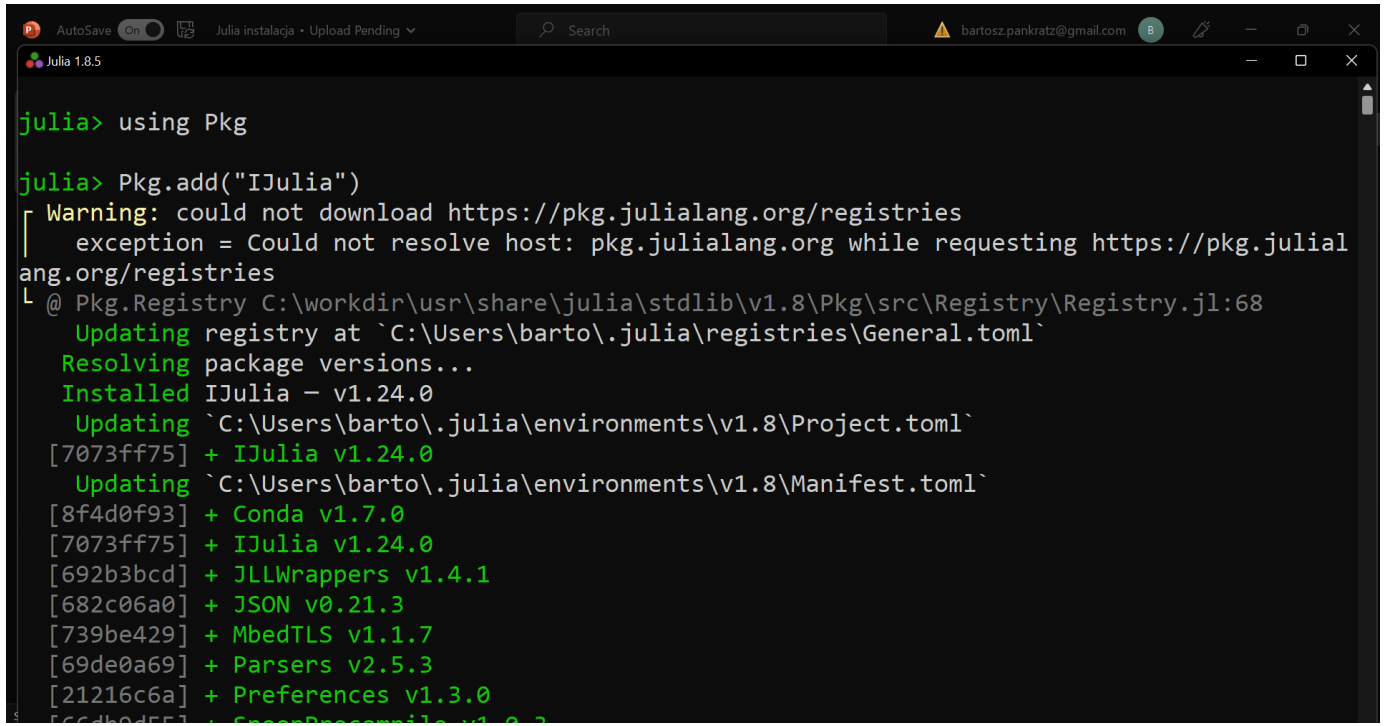
How to configure Julia?

- Firstly, we will install the necessary packages.
- It can be done in two ways:
  - or by importing the "**Pkg**" package:



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:



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-e2afdf69b7
8f\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-e2afdf69b78
f\6e47d11ea2776bc5627421d59cdcc1296c058071\build.log`
    Building PyCall → `C:\Users\barto\.julia\scratchspaces\44cfe95a-1eb2-52ea-b672-e2afdf69b78
f\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
  [91a5bcdd] 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