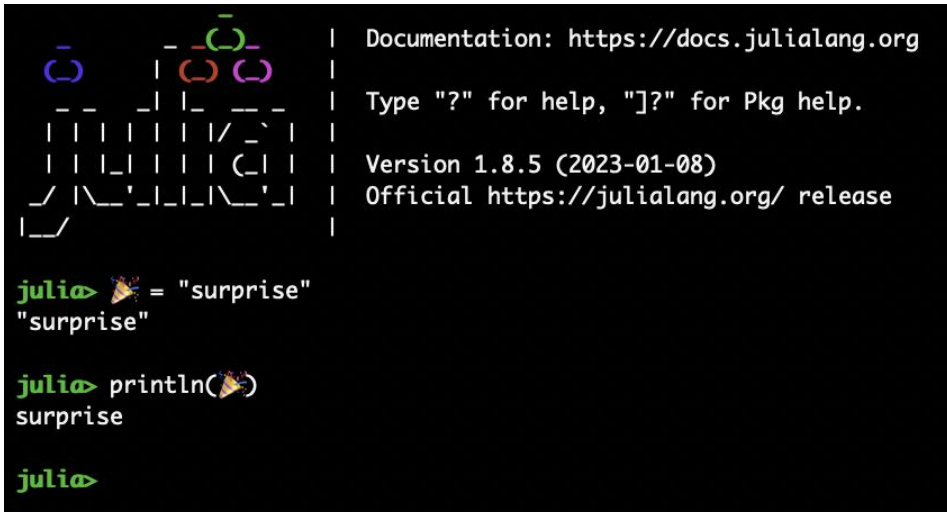
Comrade Tutorial

Warm up

Julia REPL (Read Evaluate Print Loop)

- Enter the repl with the “**julia**” command
- Enter with the local environment activated with “**julia –project**”



Julia REPL contd.

- **Julian Mode:** `julia>`(Default)
  - Use tab completion
    - `julia>` \pi<tab>
    - `julia>` prin<tab>
  - **print** something
    - `julia>` println("_____")
  - **import Plots** with "**import**" and **Comrade** with "**using**"
    - `julia>` import Plots
    - `julia>` using Comrade
- **Pkg Mode:** `pkg>`"]"
  - **activate** global environment
    - `pkg>` activate
  - **activate** local environment
    - `pkg>` activate .
  - Check project status(**st / status**)
    - `pkg>` status
    - `pkg>` st

Julia REPL contd.

- **Shell Mode:** `shell>` “;”
  - check current directory
    - `shell>` pwd
- **Search Mode:** `(reverse-i-search)``':` “ctrl+r”
  - search for your what was previously printed
    - `(reverse-i-search)`prin':`
- **Help Mode:** `help?>` “?”
  - Query Comrade
    - `help?>` Comrade
  - Query “AbstractModel”
    - `help?>` Comrade.AbstractModel

Comrade in the REPL

- Find the subtypes of AbstractModel
  - `julia> subtypes(Comrade.AbstractModel)`
- Check What geometric models exist
  - `julia> subtypes(Comrade.GeometricModel)`
- Plot a `Gaussian`
  - `julia> Plots.plot(Gaussian())`

Comrade in the REPL contd.

- Define a model **composed** of a Gaussian and MRing
  - `gauss = stretched(Gaussian(), μas2rad(10), μas2rad(10)) # Make a  10 μas Gaussian`
  - `mring = smoothed(stretched(MRing([0.5, 0.1], [0.0, 0.2]), μas2rad(20), μas2rad(20)), μas2rad(5)) # Make a MRing`
  - `model = 2mring + shifted(gauss, μas2rad(50), 0.0) #shift gaussian East by 50 μas and add its flux to mring model. Double the flux of the mring`
  -
- Plot your model
- Save your image as a .fits file
  - **julia>** modelfov = μas2rad(200)
  - **julia>** sze = 200
  - **julia>** img = intensitymap(model, modelfov, modelfov, sze, sze) #Create an intenisty map from model
  - **julia>** fits_file = joinpath((@__DIR__), "model.fits")
  - **julia>** Comrade.save(fits_file, img) #Save image as fits file

EHTImaging interface

- Load EHTImaging interface
  - `julia> load_ehtim()`
- Import fits file
  - `julia> img = ehtim.image.load_fits(fits_file)`
- Import python plotting interface and plot
  - `julia> import PythonPlot`
  - `julia> img.display()`

Geometric Modelling Example

- GeometricModellingTutorial.ipynb

CustomModel Example

- CustomModellingTutorial.ipynb

Polarized Imaging Tutorial

- PolarizedImagingTutorial.ipynb