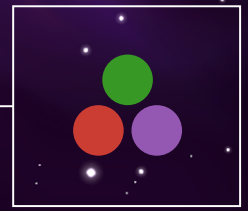

Why I use Julia
And why you might not want to
PatrickA
AstroDev

Pre-Workshop Checklist

1. **Julia** → Install Julia via JuliaUp



2. **Data** → Download the data

3. **Workspace** → Have a workspace ready for Julia coding

Julia in Brief

2009	2012	2014	2018
Work begins on Julia	Julia officially announced	First JuliaCon	Julia 1.0 released


Jeff Bezanson


Stefan Karpinski


Viral Shah


Alan Edelman

We want a language that's **open source**, with a liberal license. We want the **speed of C** with the d**ynamism of Ruby**. We want a language that's homoiconic, with **true macros like Lisp**, but with obvious, familiar **mathematical notation like Matlab**. We want something as usable for **general programming as Python**, as easy for **statistics as R**, as natural for **string processing as Perl**, as powerful for **linear algebra as Matlab**, as good at **gluing programs together as the shell**. Something that is **dirt simple to learn**, yet keeps the **most serious hackers happy**. We want it **interactive** and we want it **compiled**.

Why Julia?

JIT Compiled

Compile code once during runtime

Dynamic Types

Types are optional
More info = Faster Code

Multiple Dispatch

Overload functions
Most specific match runs

Metaprogramming

Everything in Julia has a type
Code which writes code

Opinionated

Built in:
Package Management, Profiling, Debugging, Logging

Extendable

Robust Package Ecosystems: Visualisation, Statistics, Machine Learning, Quantum Computing, HPC, and **Astronomy!**

Interactable

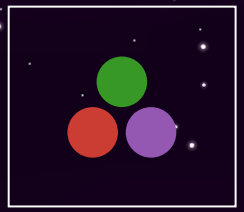
Interface with C, Fortran, C++, Python, R, Java, Mathematica, Matlab, etc...

Basics: Interacting with Julia

REPL	Notebook	Scripts
Interactive command-line *read-eval-print loop*	Reactive and Reproducible	*julia script.jl* *include("script.jl")*
● *]* - Package Mode ● ? - Help Mode ● ; - Shell Mode ● Colours ● Tab Completion ● Unicode / LaTeX Input	● Automatic Cell Re-evaluation ● *PlutoUI* – Sliders, Buttons, etc... ● Built-in Package Manager ● No mutable workspace, no hidden state	● Feature-Rich VSCode Integration ● Emacs, Vim, Atom, Zed, Notepad++, Sublime, Intellij, and more!
julia ≅ ipython	***Pluto ≅ Jupyter***	***script.jl ≅ script.py***

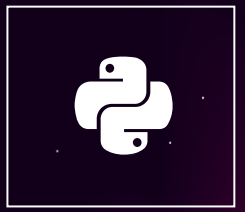
Basics: Reading, analysing, and writing

JuliaWorkshop/Code/1. Basics/Scripts



Julia

julia script.jl
output_jl.txt

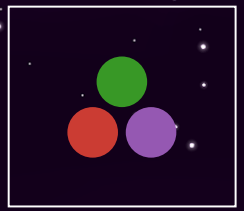


Python

python script.py
output_py.txt

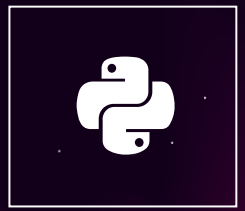
Basics: Environments, Workflow, and Plotting

JuliaWorkshop/Code/1. Basics/Environments/AnalyseLightcurve.{jl,py}



Julia

julia script.jl
output_jl.txt



Python

python script.py
output_py.txt

Basics: Environments and Plotting

] generate AnalyseLightcurve.jl ←	Create Environment	→ *python -m venv .venv*
] activate . ←	Activate Environment	→ *source .venv/bin/activate*
using AnalyseLightcurve ←	Import Project	→ *import AnalyseLightcurve*
AnalyseLightcurve.main() ←	Run Project	→ *AnalyseLightcurve.main()*

Basics: Iterative Workflow

using Revise ← Activate AutoReload → *%load_ext autoreload*
%autoreload 2

edit(AnalyseLightcurve) ← Edit code → *%edit AnalyseLightcurve.py*

;imv Supernova\ Lightcurve.svg ← View results → *!imv Supernova\ Lightcurve.svg*