



## Julia Workshop

Carsten Bauer @ University of Cologne, March 2021

	Monday	Tuesday	Wednesday	Thursday
9:30 – 11:00 (1.5h)	Types + Dispatch	Performance	Machine Learning (Physics)	
	Short break			
11:15 – 12:15 (1h)	Custom types	Gotchas	Parallel	Hackathon
	Lunch break			
13:45 – 14:45 (1h)	Abstraction	Linear Algebra	Computing	
	Short break			
15:00 – 16:00 (1h)	Specialization	Automatic Differentiation	Q&A	

# juliacan

2018





# MeetAnyway

Hackathon @ Advanced Julia V

meetanyway.com/events/julia-workshop-hackathon

Carsten Bauer

Edit event

Thursday Mar 18, 2021 at  
09:00 AM - 05:00 PM  
Europe/Berlin

Add to Calendar

Enter event space

Share Event ↗



## Hackathon @ Advanced Julia Workshop

Organized by ML4Q

### Details

A hackathon is a social programming event.

The idea is to bring people together so that they can work on certain projects either

# GitHub repository

<https://github.com/crstnbr/JuliaCologne21>

The screenshot shows a GitHub repository page for 'crstnbr/JuliaCologne21'. The repository has 1 branch and 0 tags. It contains 35 commits from 'crstnbr' with commit IDs like 77c994b and dates ranging from 2 days ago to 6 days ago. The files listed include Day1, Day2, Day3, backup, orga, .gitignore, LICENSE, Manifest.toml, Project.toml, README.md, and RESOURCES.md. The repository has 1 unwatcher, 3 stars, and 0 forks. The 'About' section describes it as a 'Julia workshop for undergrad physicists' with links to Readme and MIT License. There are sections for Releases (no releases published), Packages (no packages published), and Languages (not shown).

crstnbr / JuliaCologne21

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

Go to file Add file Code

File	Commit Message	Time Ago
Day1	point	3 days ago
Day2	minor	4 days ago
Day3	minor	2 days ago
backup	minor	2 days ago
orga	minor	2 days ago
.gitignore	blub	3 months ago
LICENSE	blub	3 months ago
Manifest.toml	improvements	4 days ago
Project.toml	improvements	4 days ago
README.md	Update README.md	2 days ago
RESOURCES.md	reinit	6 days ago

About

Julia workshop for undergrad physicists

Readme

MIT License

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

# GitHub repository

!Sorry!

JuliaNRW21?!

<https://github.com/crstnbr/JuliaCologne21>



The image shows two side-by-side GitHub repository pages:

**Left Repository: JuliaNRW21**

- Code:** master branch, 2 branches, 0 tags.
- Commits:**
  - crstnbr Merge branch 'master' of github.com:crstnbr/JuliaNRW21 (9d9796d, 9 days ago)
  - Day1 Fix links to docs.julialang.org
  - Day2 Fix links to docs.julialang.org
  - Day3 renaming
  - backup Fix links to docs.julialang.org
  - orga install.jl
  - .gitignore init
  - LICENSE init
  - Manifest.toml rc1
  - Project.toml Add WebIO for examples in Day1/5\_generic\_programming.ipynb
  - README.md Update README.md
  - RESOURCES.md resources
  - install.jl Add WebIO for examples in Day1/5\_generic\_programming.ipynb
- README.md:** Julia Workshop - HPC.NRW

**Right Repository: JuliaCologne21**

- Code:** master branch, 1 branch, 0 tags.
- Commits:**
  - crstnbr minor (77c994b, 2 days ago)
    - Day1 point (3 days ago)
    - Day2 minor (4 days ago)
    - Day3 minor (2 days ago)
    - backup minor (2 days ago)
    - orga minor (2 days ago)
    - .gitignore blub (3 months ago)
    - LICENSE blub (3 months ago)
    - Manifest.toml improvements (4 days ago)
    - Project.toml improvements (4 days ago)
    - README.md Update README.md (2 days ago)
    - RESOURCES.md reinit (6 days ago)
    - install.jl day 1 overhaul (5 days ago)
- About:** Julia workshop for undergrad physicists
- Readme:**
- MIT License:**
- Releases:** No releases published. Create a new release.
- Packages:** No packages published. Publish your first package.
- Languages:** Jupyter Notebook 100.0%

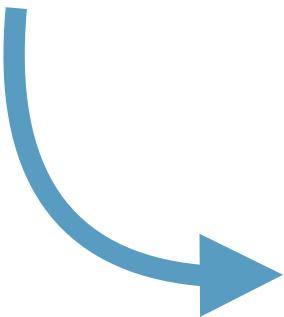
## GitHub repository

<https://github.com/crstnbr/JuliaCologne21>

!Sorry!  
JuliaNRW21?!



```
    ] update WorkshopWizard  
  
    using WorkshopWizard  
    WorkshopWizard.run_wizard()
```



- Takes minutes to start
- Slow machines
- **Data not persistent!**

A screenshot of a Jupyter notebook interface. The title bar shows 'JuliaCologne21' and the URL 'hub-binder.mybinder.ovh/user/crstnbr-juliacologne21-binder-2moyig6s/tree/JuliaCologne21'. The main area displays a file tree under 'JuliaCologne21'. The files listed are:

Name	Last Modified	File size
0	seconds ago	
backup	seconds ago	
Day1	seconds ago	
Day2	seconds ago	
Day3	seconds ago	
orga	seconds ago	
install.jl	seconds ago	549 B
LICENSE	seconds ago	1.07 kB
Manifest.toml	seconds ago	49.1 kB
Project.toml	seconds ago	1.22 kB
README.md	seconds ago	3.6 kB
RESOURCES.md	seconds ago	1.61 kB

# Mentimeter poll...



# The Power of Language

## Vandermonde matrix

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{bmatrix}$$

**vander(x)**

`numpy.vander(x)`

# Python

```
def vander(x, N=None, increasing=False):
    x = asarray(x)
    if x.ndim != 1:
        raise ValueError("x must be a one-dimensional array or sequence.")
    if N is None:
        N = len(x)

    v = empty((len(x), N), dtype=promote_types(x.dtype, int))
    tmp = v[:, ::-1] if not increasing else v

    if N > 0:
        tmp[:, 0] = 1
    if N > 1:
        tmp[:, 1:] = x[:, None]
        multiply.accumulate(tmp[:, 1:], out=tmp[:, 1:], axis=1)

    return v
```

calls  
r sequence.")

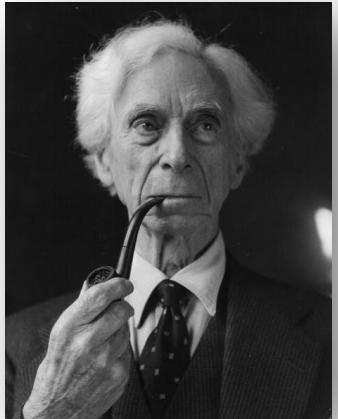
# C template

```
/* Take a reference to our file manager */  
uses
```

# Julia

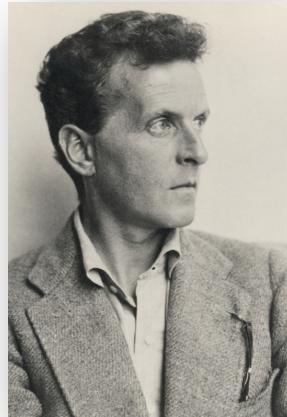
```
function vander(x::AbstractVector{T}) where T
    m = length(x)
    V = Matrix{T}(undef, m, m)
    for j = 1:m
        V[j,1] = one(x[j])
    end
    for i= 2:m
        for j = 1:m
            V[j,i] = x[j] * V[j,i-1]
        end
    end
    return V
end
```

# The Power of Language



Language serves not only to express thoughts, but to make possible thoughts which could not exist without it.

**Bertrand Russell**



The limits of my language mean the limits of my world.

**Ludwig Wittgenstein**

When language has been well chosen, one is astonished... (in mathematics what's possible...)

**Henri Poincaré**





What does science need from a  
programming language?

**Performance!**

Workflow: roll up sleeves & performance engineer



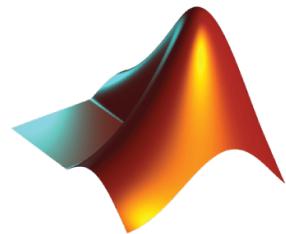
What does science need from a  
programming language?

Easy to write and read !

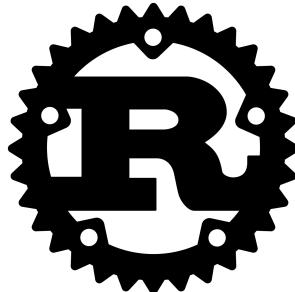
Fast and scalable !

Interactive !

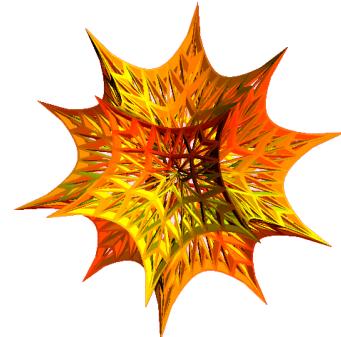
# There's a plethora of programming languages



MATLAB



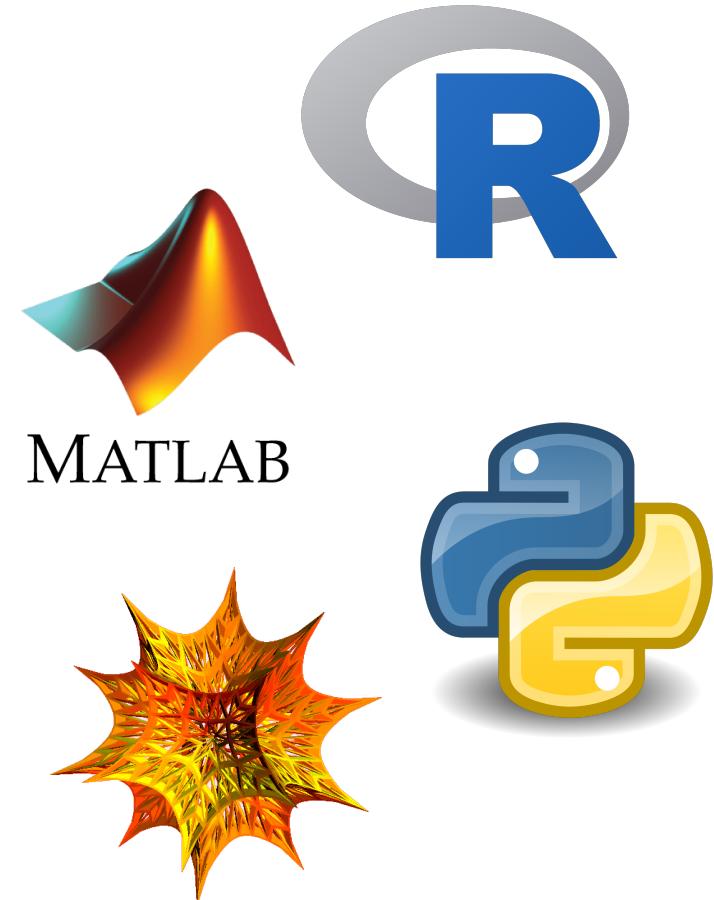
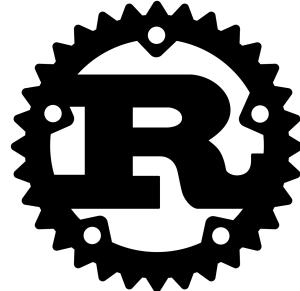
Fortran



# There's a plethora of programming languages



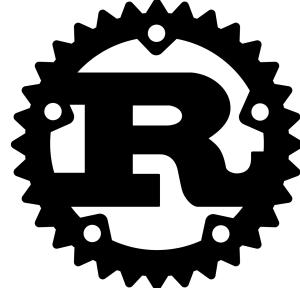
Fortran



# There's a plethora of programming languages

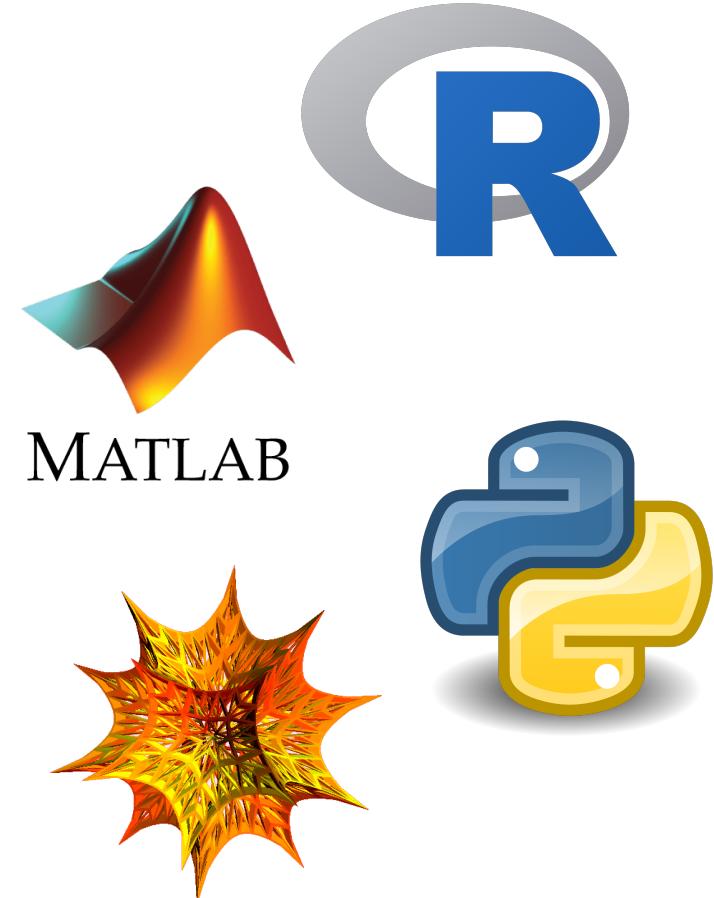


Fortran



Speed

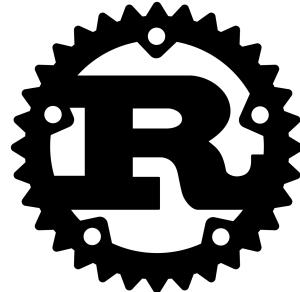
Convenience



# There's a plethora of programming languages

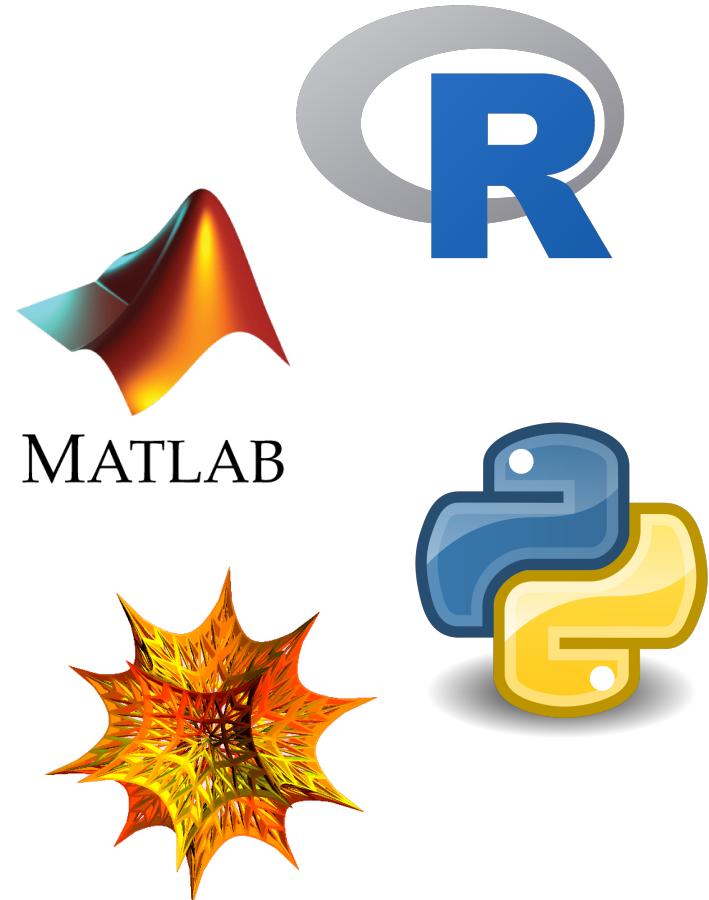


Fortran



Compiled

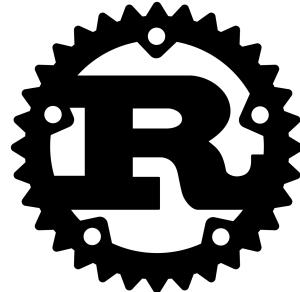
Interpreted



# There's a plethora of programming languages

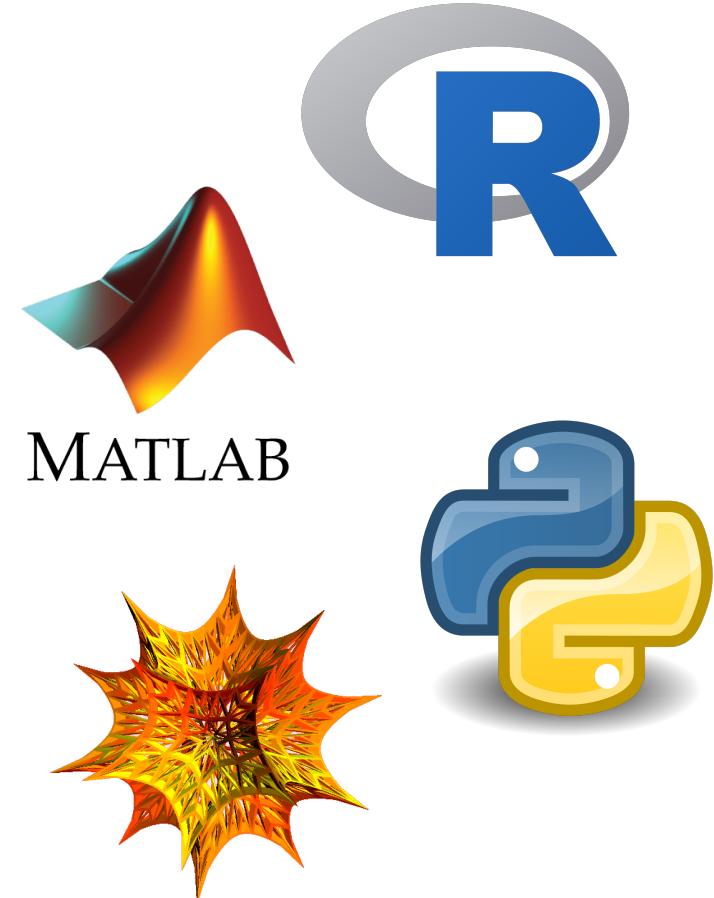


Fortran



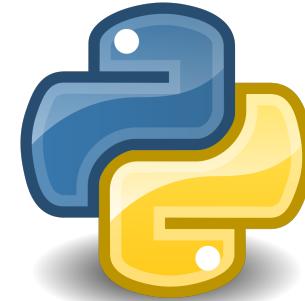
Static

Dynamic



# The “two language problem”

a.k.a Ousterhout's dichotomy



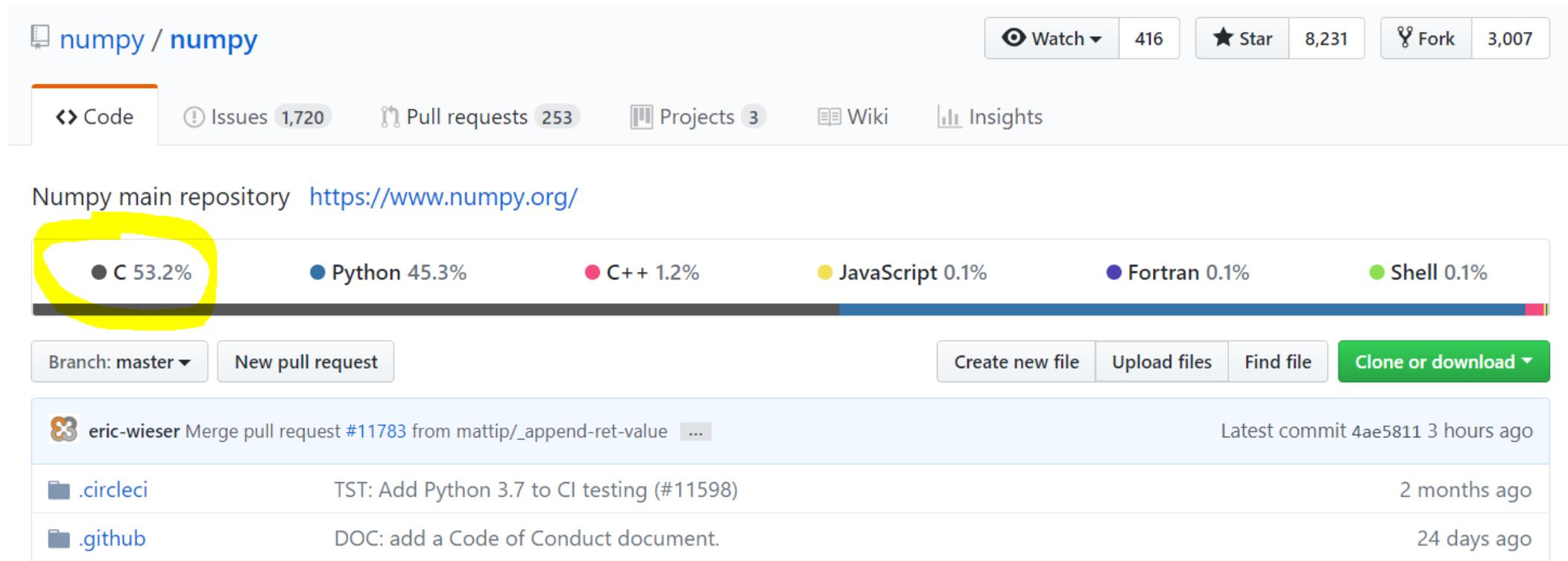
Prototype

---

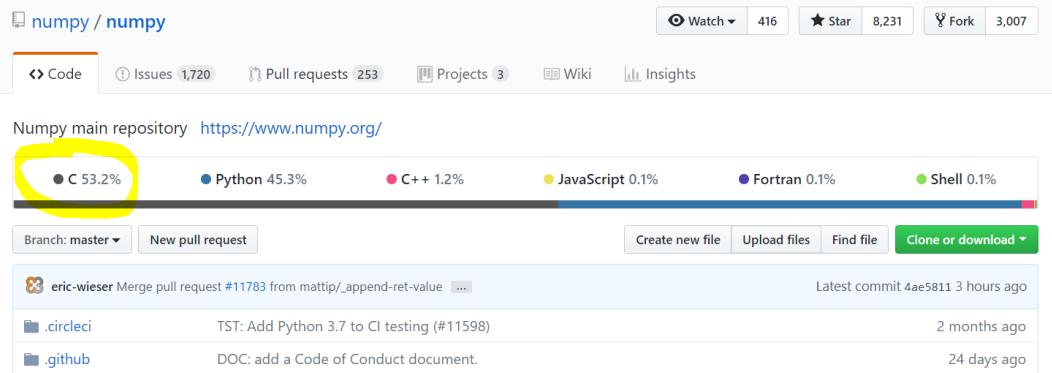
Production



# The “two language problem”



# The “two language problem”



Developer

User



# The “two language problem”

static	dynamic
compiled	interpreted
user types	standard types
standalone	glue



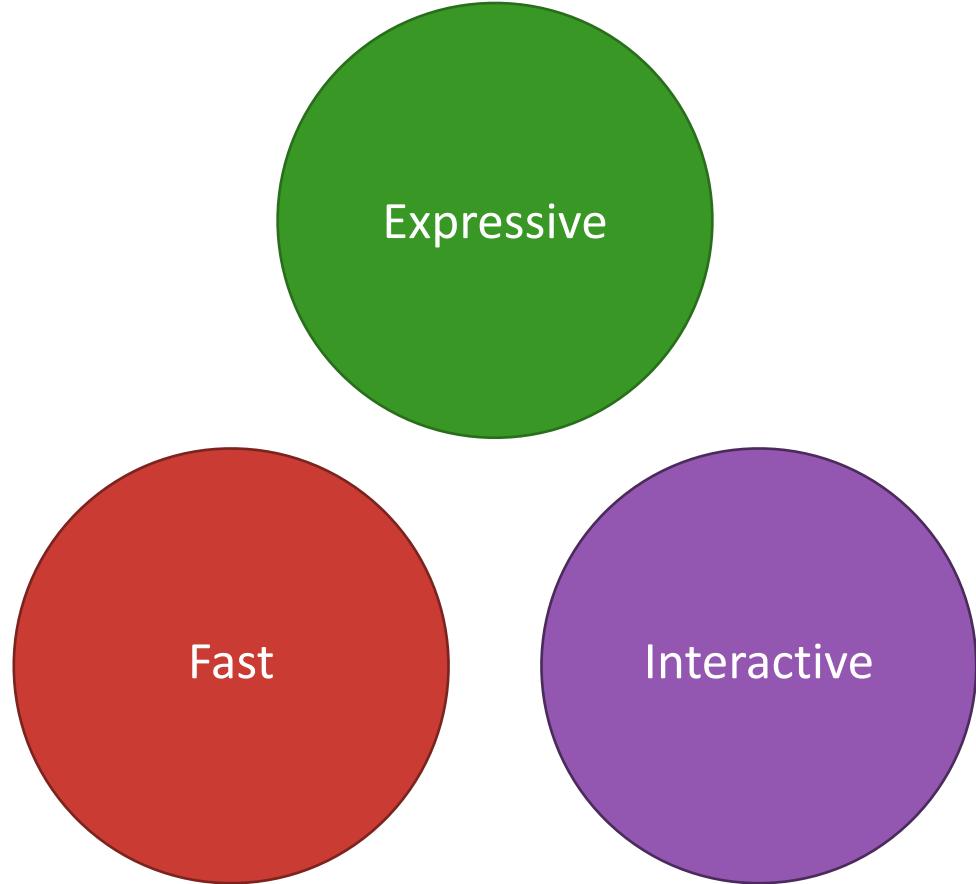
dynamic

compiled

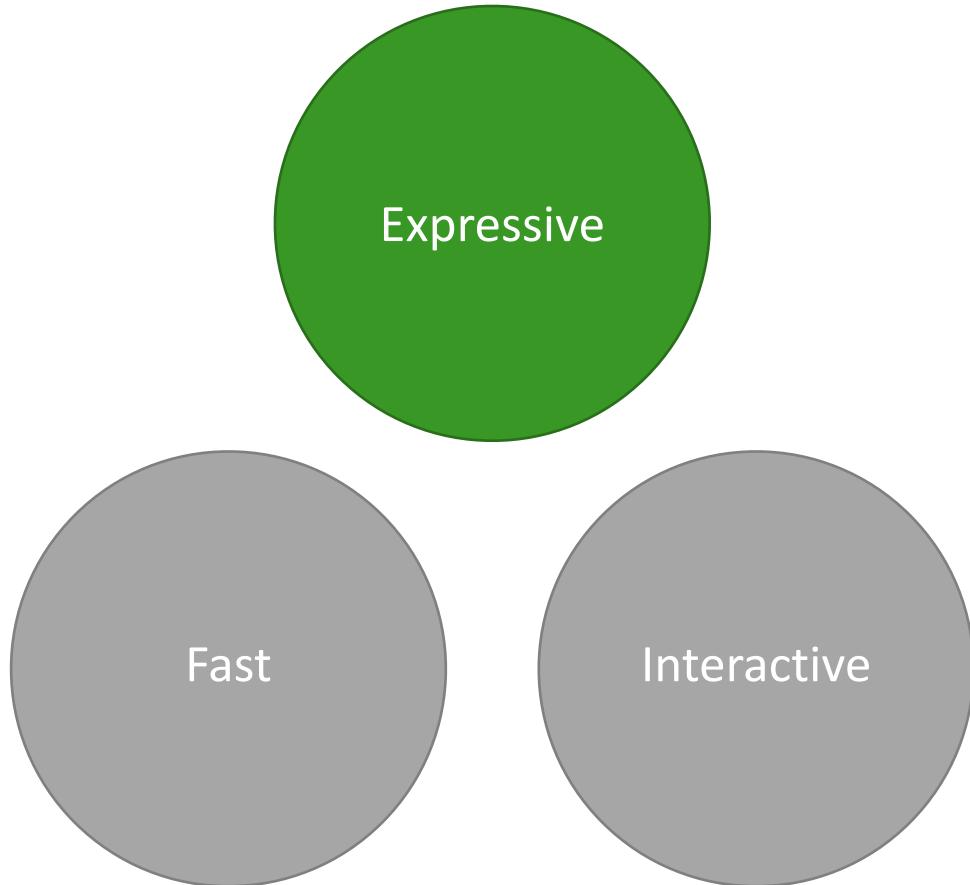
user types **and** standard types

standalone **or** glue

# The unification



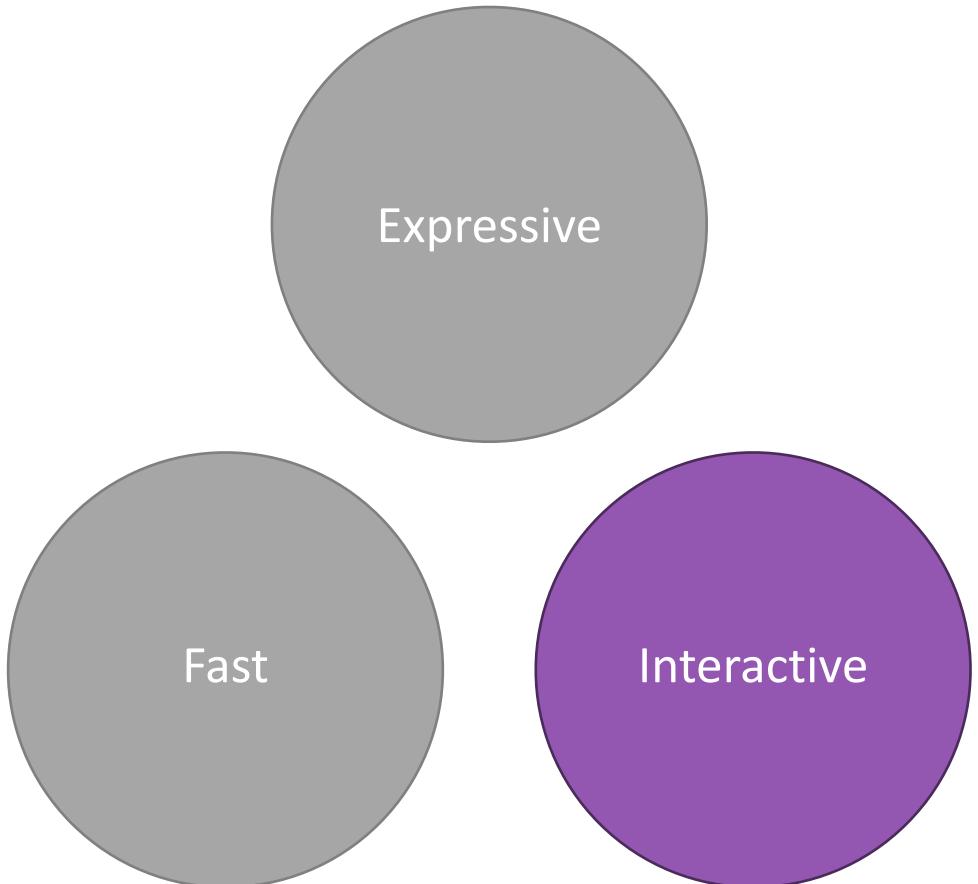
# The unification



```
function babylonian(a; N = 10)
    @assert a > 0 "a must be > 0"
    t = (1+a)/2
    for i = 2:N
        t = (t + a/t)/2
    end
    t
end

babylonian(π) ≈ √π
```

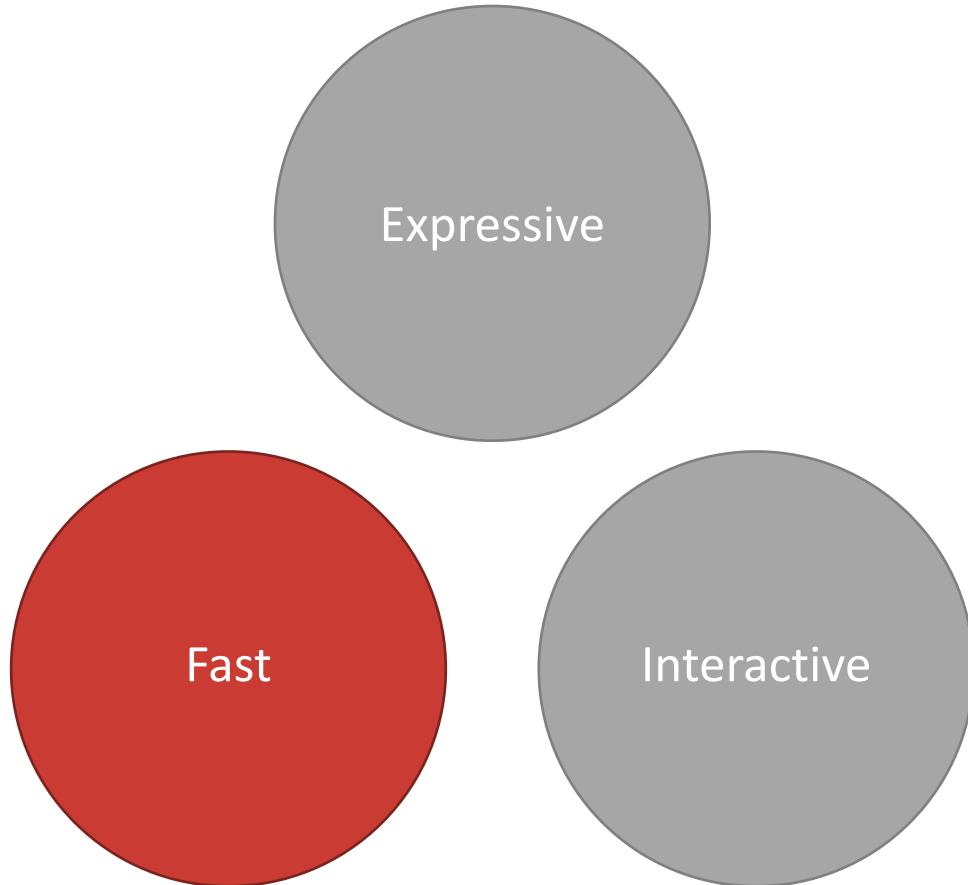
# The julia unification



A screenshot of the Julia IDE interface. On the left, the Project pane shows a file named `profile_test.jl` with code related to FFTW and matrix operations. In the center, the Main pane displays the code and its execution results. On the right, there are two panes: one for "Plots" showing several line graphs, and another for "profiler" showing memory usage over time.

A screenshot of a Jupyter Notebook titled "Lorenz Differential Equations". The notebook interface includes a toolbar at the top, a code cell, and a text cell. The text cell contains a section titled "Exploring the Lorenz System" with a warning about relying on the server. Below this, it says "This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters ( $\sigma$ ,  $\beta$ ,  $\rho$ ) are varied, including what are known as chaotic solutions. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963." A code cell below uses the `interact` function to allow users to vary parameters  $\sigma$ ,  $\beta$ , and  $\rho$  and see the resulting trajectories. At the bottom, a plot shows the characteristic butterfly-shaped Lorenz attractor.

# The unification

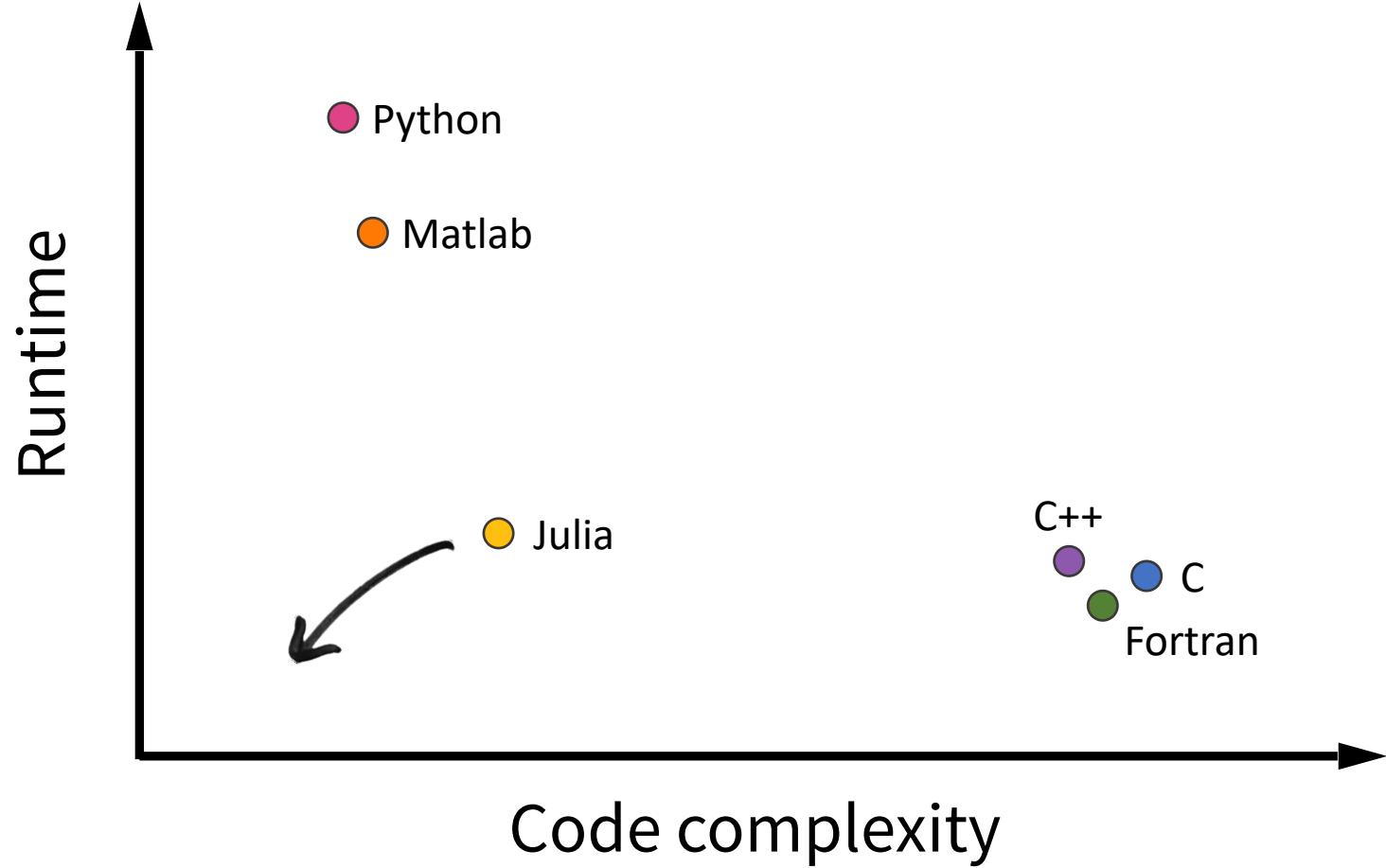


```
julia> function sumup()
           x = 0
           for i in 1:100
               x += i
           end
           x
       end
sumup (generic function with 2 methods)

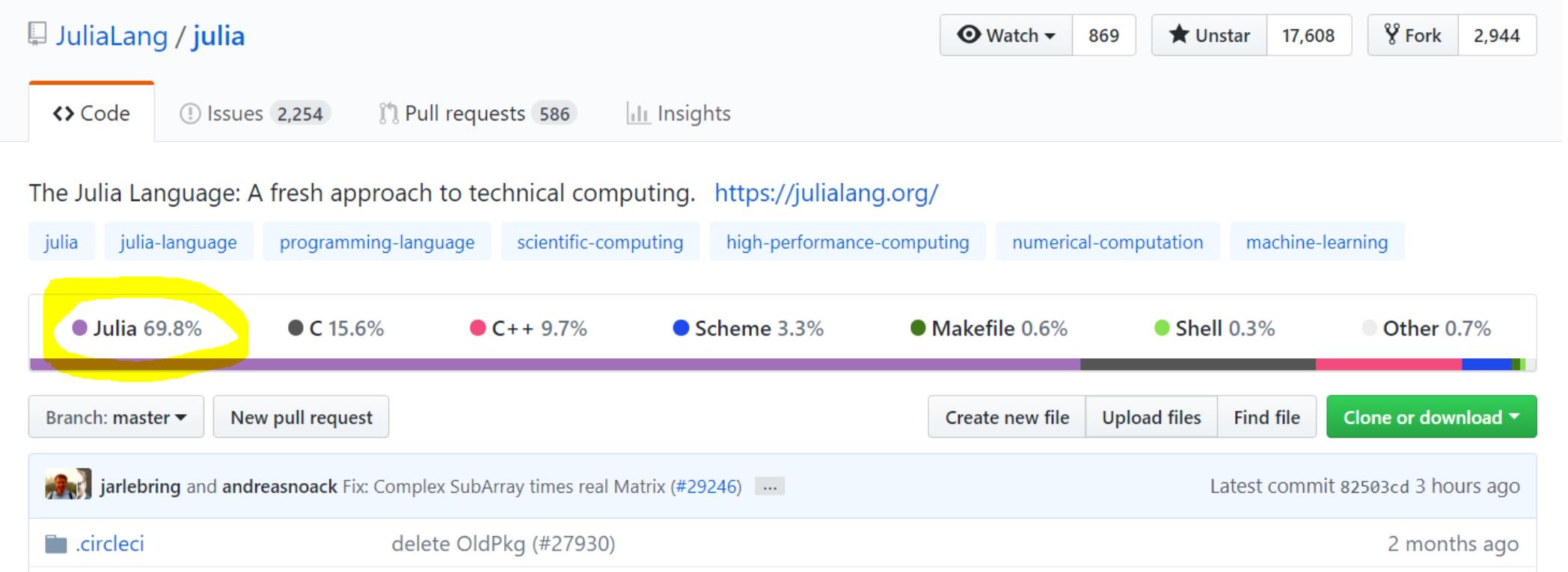
julia> @code_llvm debuginfo=:none sumup()

; Function Attrs: uwtable
define i64 @julia_sumup_12626() #0 {
top:
    ret i64 5050
}
```

Just returns the answer!



# Free and open source

A screenshot of the GitHub repository page for JuliaLang/julia. The page shows the repository's name, statistics (869 stars, 17,608 forks, 2,944 issues, 586 pull requests), and various tabs like Code, Issues, Pull requests, and Insights. A prominent feature is a horizontal bar chart showing the percentage of code by language: Julia (69.8%), C (15.6%), C++ (9.7%), Scheme (3.3%), Makefile (0.6%), Shell (0.3%), and Other (0.7%). The "Julia" entry is highlighted with a yellow oval. Below the chart, there are buttons for Branch: master, New pull request, Create new file, Upload files, Find file, and Clone or download. A recent commit from jarlebring and andreasnoack is shown, along with a note about a .circleci update.

The Julia Language: A fresh approach to technical computing. <https://julialang.org/>

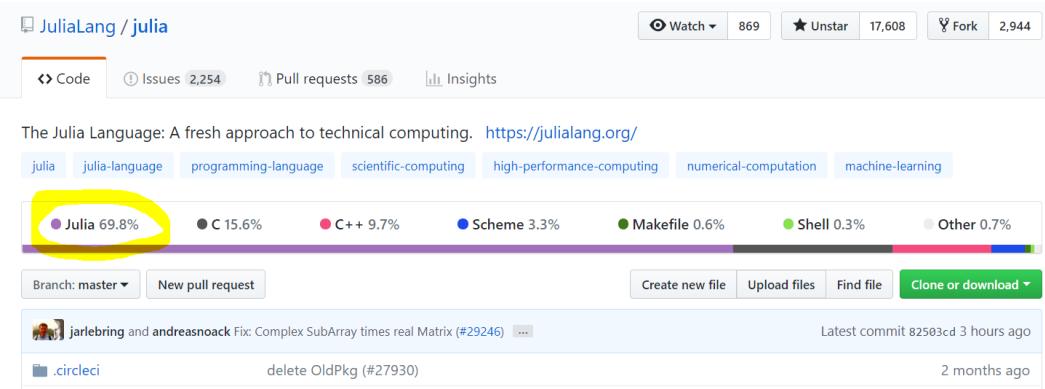
Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

Latest commit 82503cd 3 hours ago

.circleci delete OldPkg (#27930) 2 months ago

Language	Percentage
Julia	69.8%
C	15.6%
C++	9.7%
Scheme	3.3%
Makefile	0.6%
Shell	0.3%
Other	0.7%

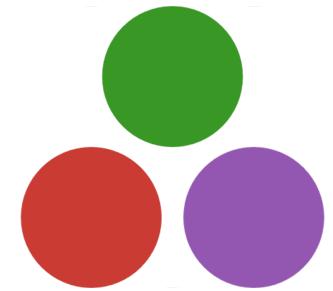
# Inviting



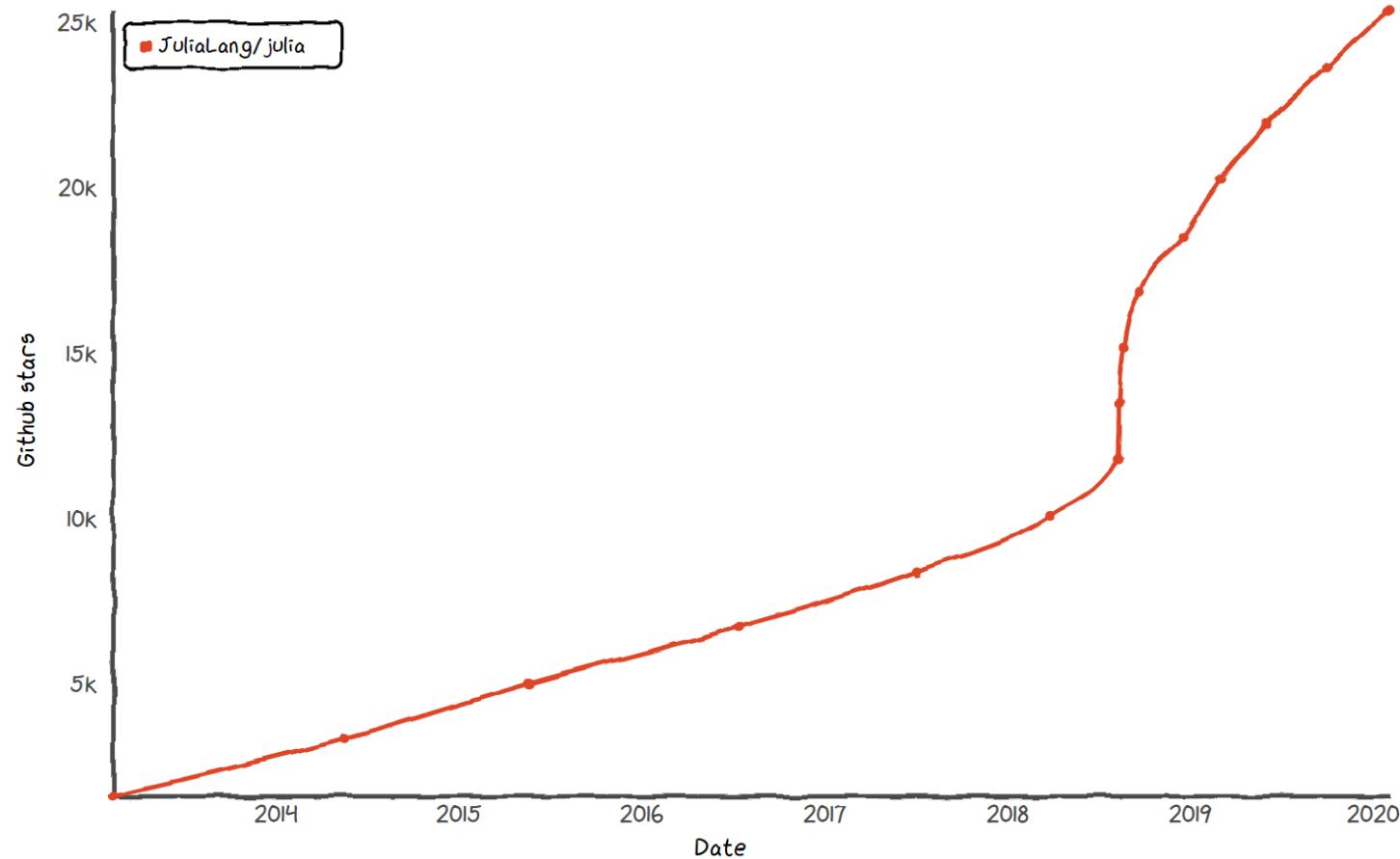
User



Developer



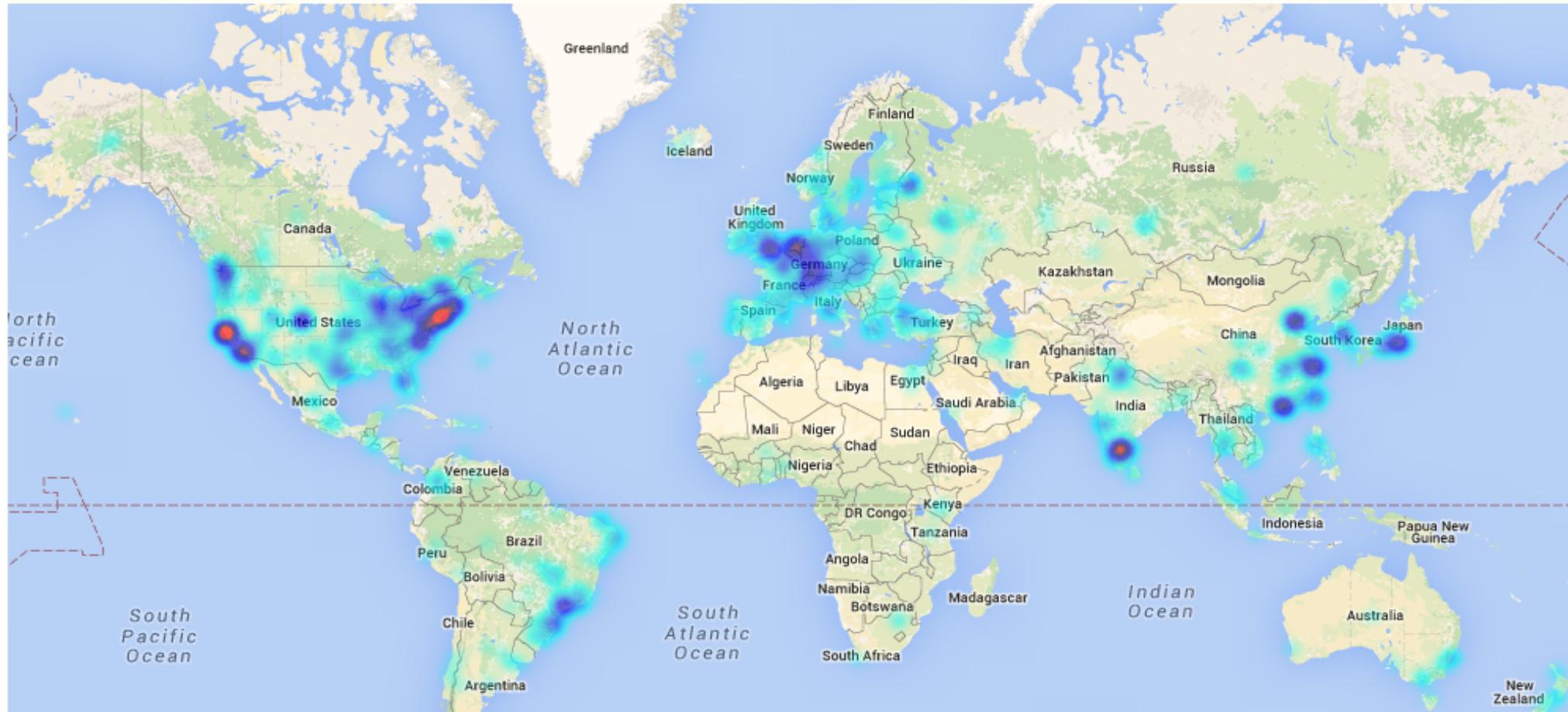
# Julia GitHub stars



\* Base language, does not include packages

# A global community

More than 25 Million downloads, >5000 packages



James H. Wilkinson Prize  
For Numerical Software

Stefan Karpinski  
Viral B. Shah  
Jeff Bezanson

(2019)



Forbes  
30 under 30

Keno Fischer

(2019)

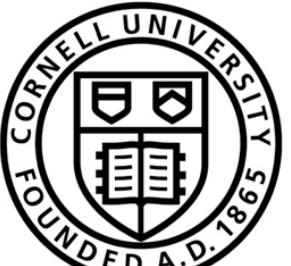


IEEE Babbage Prize  
IEEE Fellow

Prof. Alan Edelman

(2018)





EMORY  
UNIVERSITY



S  
Stanford  
University



UNIVERSITY  
of  
GLASGOW



UNIVERSITE  
PAUL  
SABATIER



Western  
UNIVERSITY · CANADA



CU THE CITY  
UNIVERSITY  
OF  
NEW YORK

EPFL  
ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE



M Ű E G Y E T E M 1 7 8 2



TOKYO METROPOLITAN UNIVERSITY  
首都大学東京



UCLA

AGH



Let's get started!