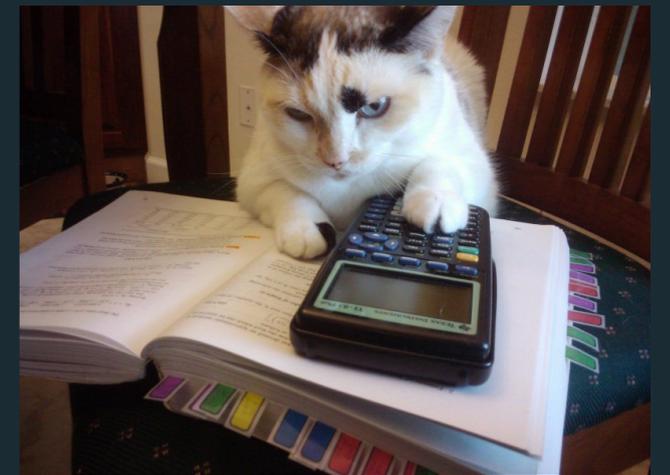
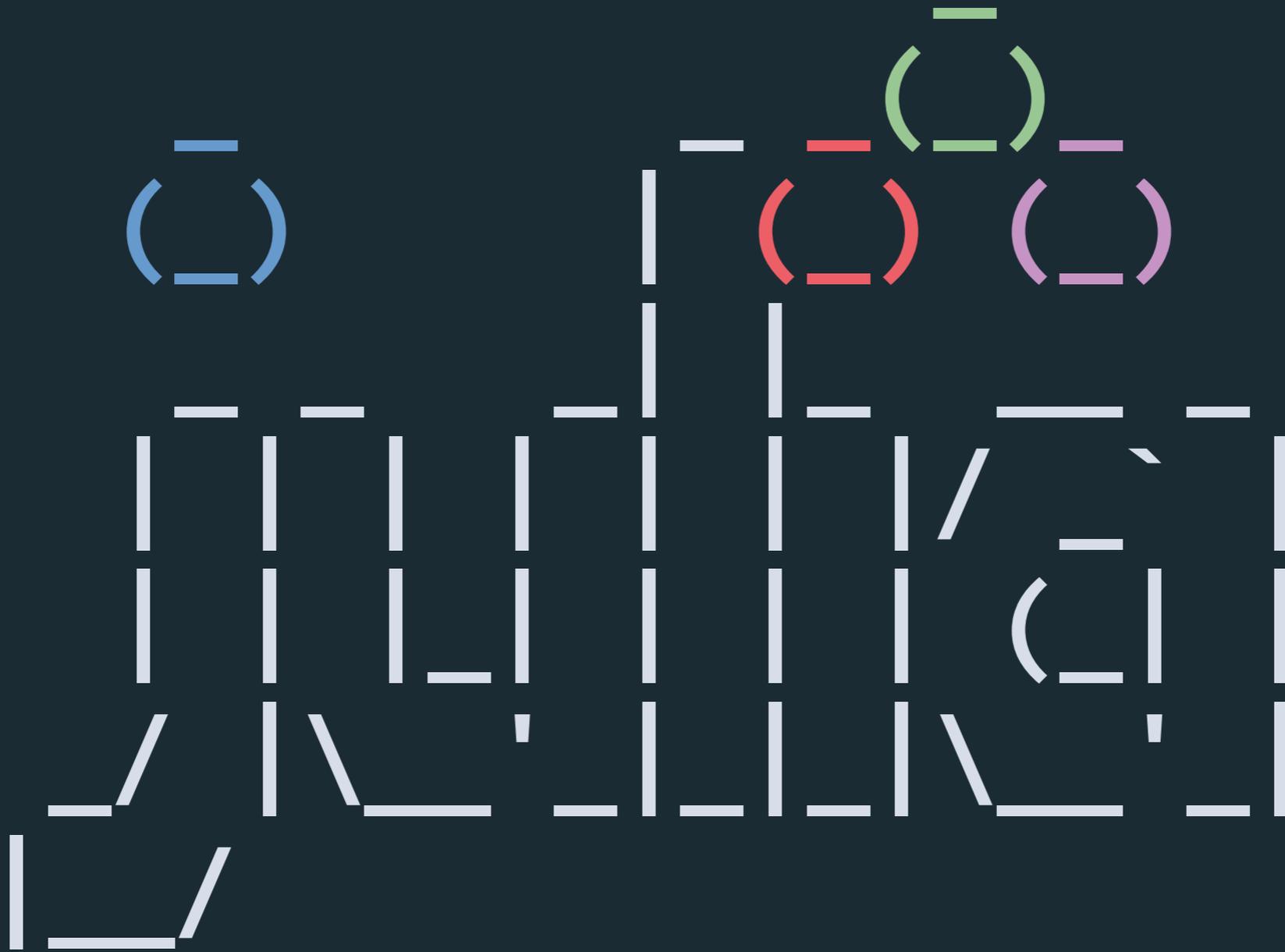


, &



Sacha Verweij and Jane Herriman



the two language problem

productive
(lisp, python, ruby, matlab)

performant
(asm, fortran, c, c++)

either write code efficiently ... or write efficient code

typical workaround?

use two languages
(prototype + production)

ergo *the two language problem*

“looks like python, feels like lisp, runs like c”

...looks like python...

python

```
def sum(a):  
    s = 0.0  
    for x in a:  
        s += x  
    return s
```

julia

```
function sum(a)  
    s = 0.0  
    for x in a  
        s += x  
    end  
    return s  
end
```

...feels like Lisp...

homoiconic

dynamic

parametric

multiple dispatch

highly polymorphic

...

...runs like c...

python

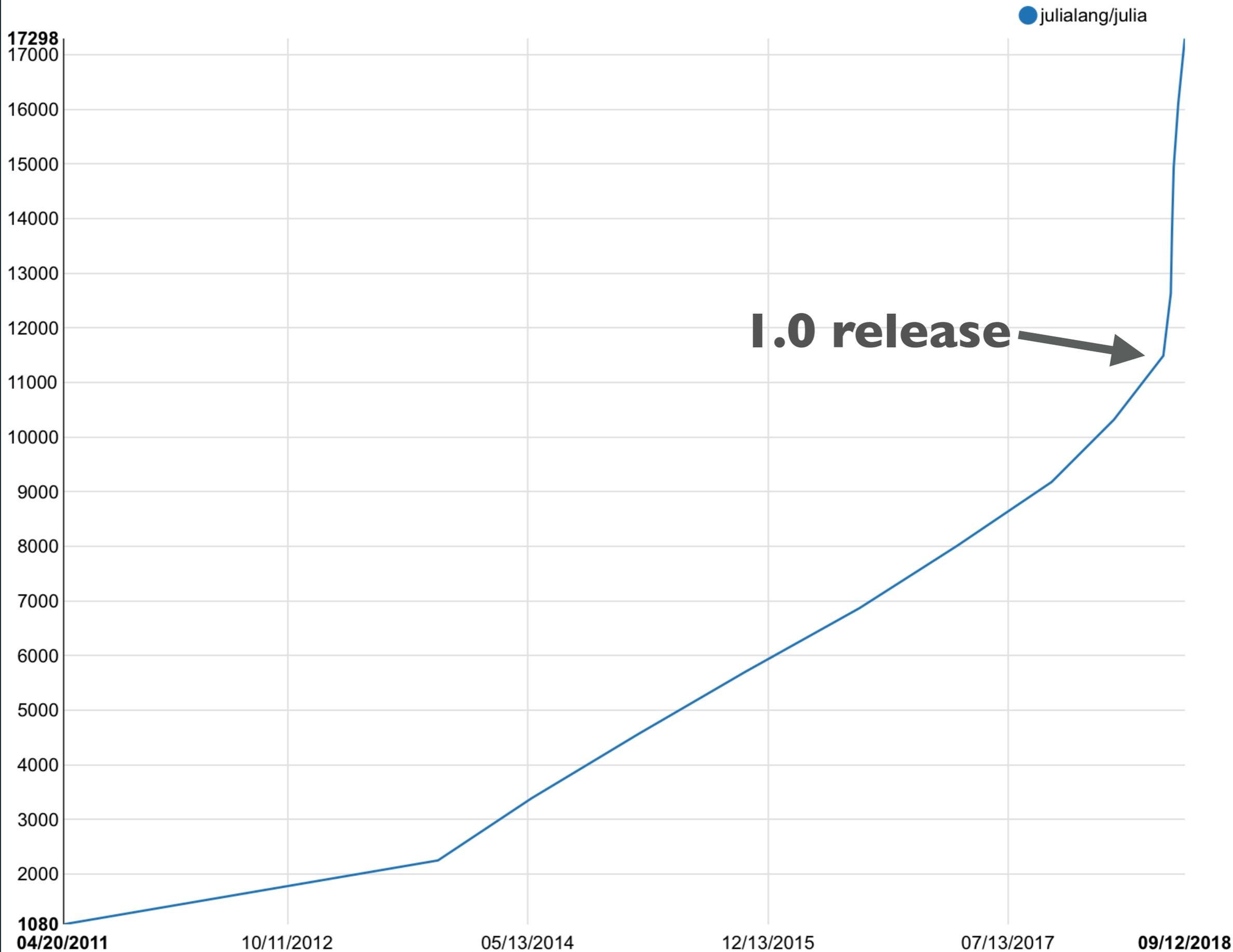
```
def sum(a):  
    s = 0.0  
    for x in a:  
        s += x  
    return s
```

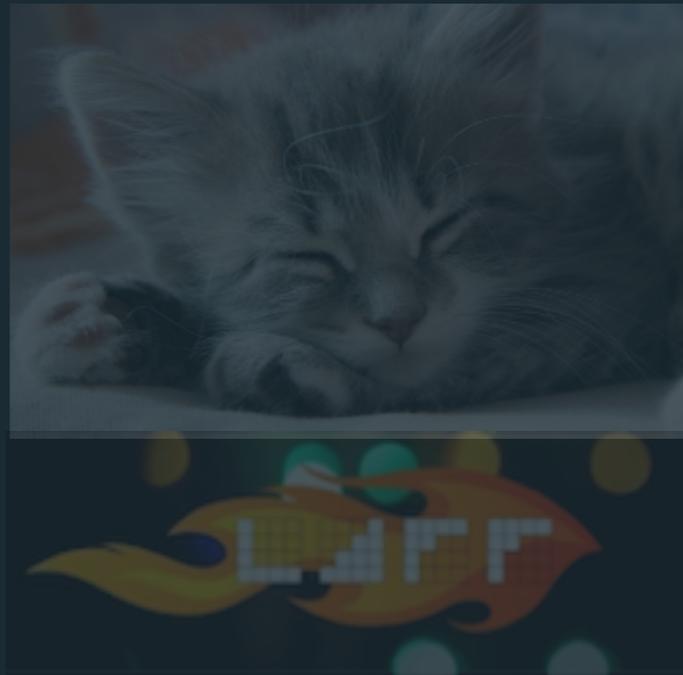
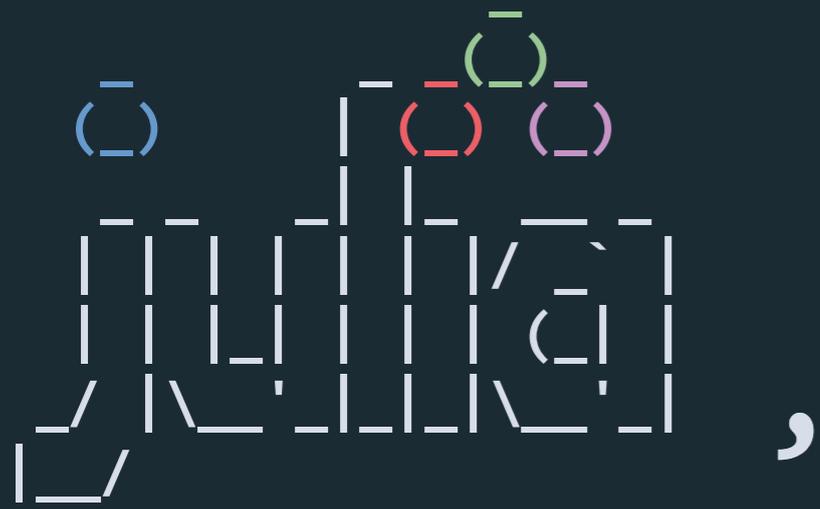
julia

```
function sum(a)  
    s = 0.0  
    for x in a  
        s += x  
    end  
    return s  
end
```

“looks like python, feels like lisp, runs like c”

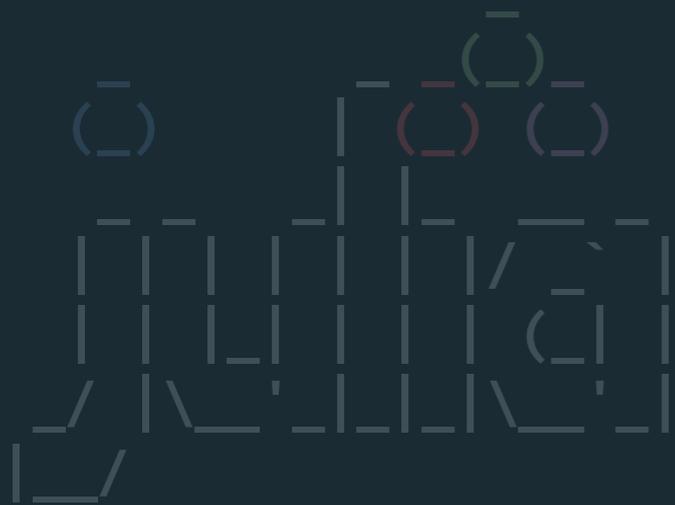
github stars



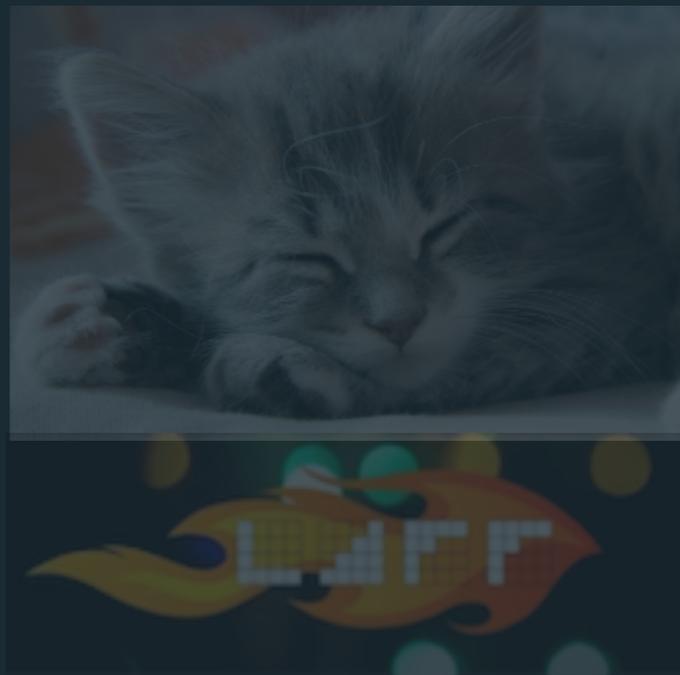


, &

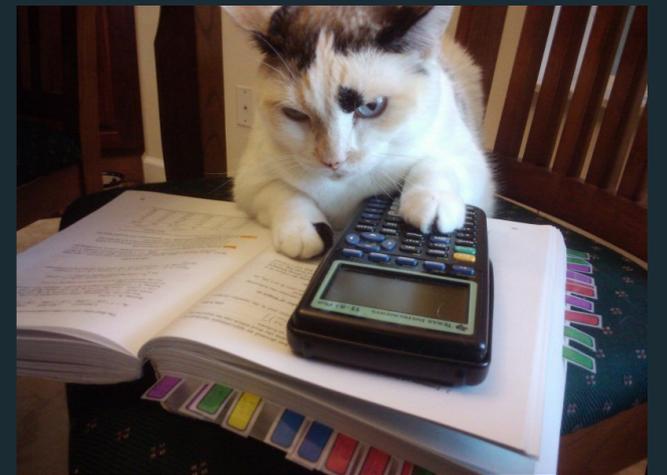




,



, &



an two language problem for teaching?

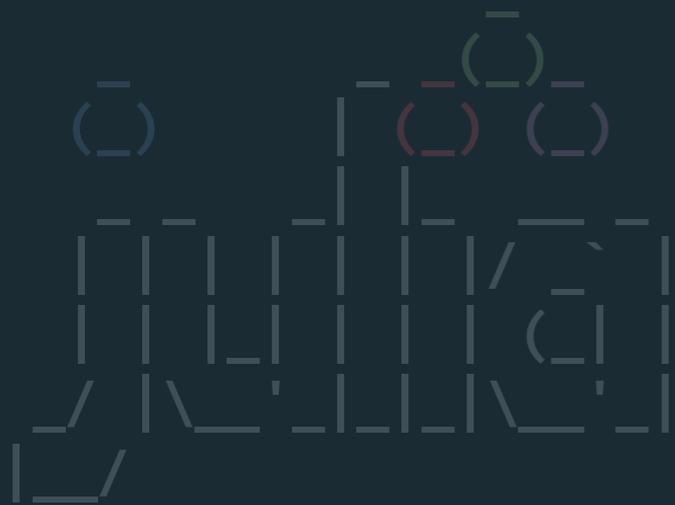
accessible & productive
(python, ruby, lua, matlab, ...)

powerful &/| performant
(c/c++, lisps, fortran, rust, ...)

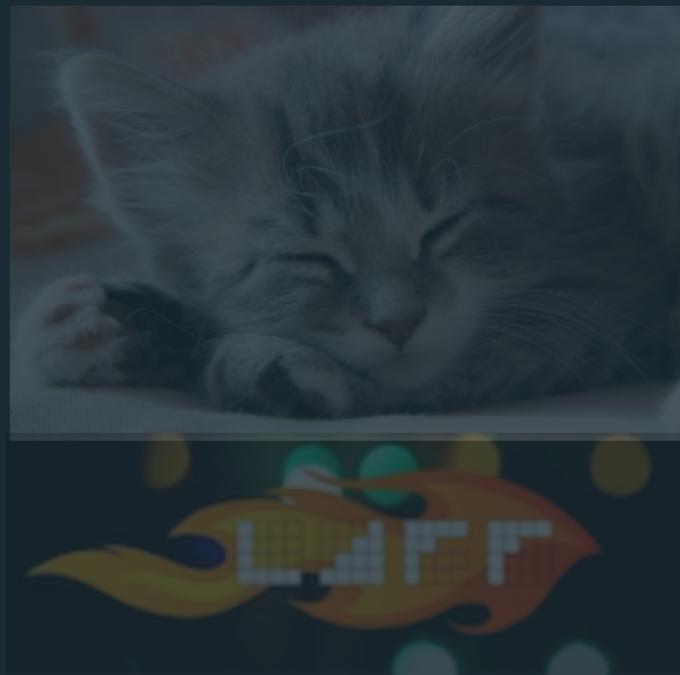
intro CS
(loops, variables,
functions, control flow, ...)

advanced CS
(types, architectures,
compilers, performance, ...)

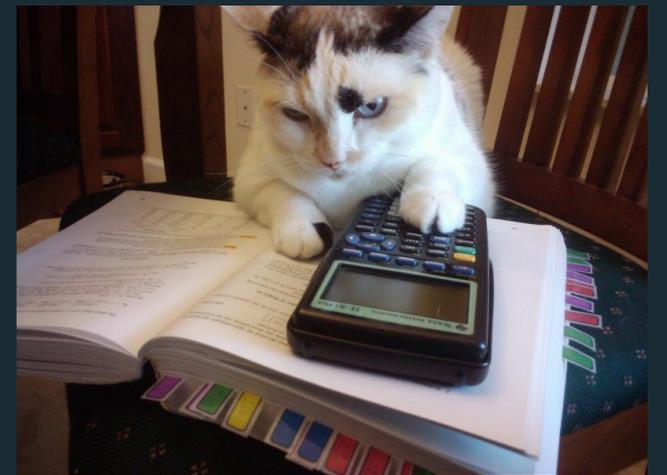
we can smooth the way with 

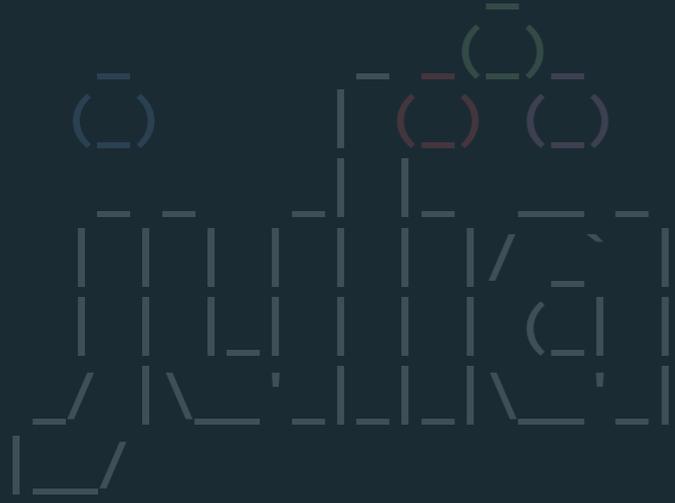


,



, &



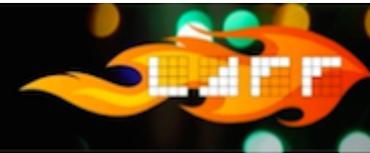


,



, &



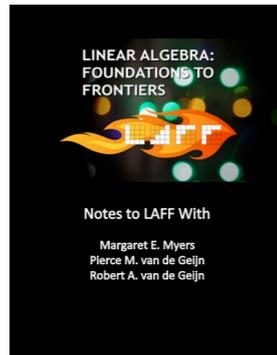


Linear Algebra: Foundations to Frontiers (LAFF)

A MOOC on the [edX platform](#).

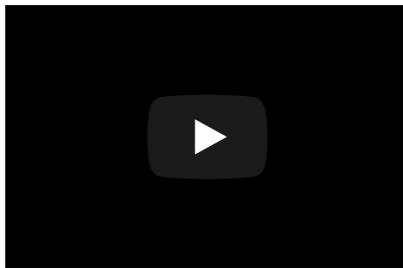
[Current offering Fall 2018 \(Aug. 1, 2018 - Dec. 17, 2018\)](#)
[Reviews](#)

Download! [Linear Algebra: Foundations to Frontiers - Notes to LAFF With](#)



An e-book (PDF) that integrates the materials for LAFF, including 270+ short videos, answers to exercises, browser-based activities, and programming exercises for MATLAB.

Watch the "Sizzle" Video":



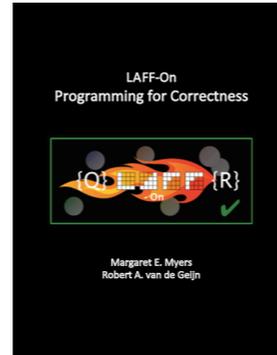
Funded in part by the [University of Texas System](#) and the [National Science Foundation](#) (grant ACI-1148125).

LAFF-On Programming for Correctness (LAFF-On)

A MOOC on the [edX platform](#).

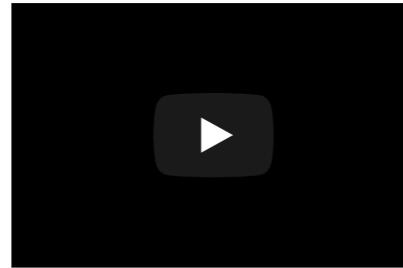
[Offering that started May 15, 2018 is now archived.](#)

Download! [LAFF-On Programming for Correctness](#) (Complete notes with answers.)



An e-book (PDF) that integrates the materials for LAFF-On, including short videos, answers to exercises, browser-based activities, and programming exercises for MATLAB.

Watch the "Sizzle" Video":



Funded in part by a gift from [MathWorks](#) and the [National Science Foundation](#) (grant ACI-1550493).

LAFF-On Programming for High Performance (LAFF-On PfHP)

Materials under development.

Download! [LAFF-On Programming for High Performance](#) (Notes and materials under development)



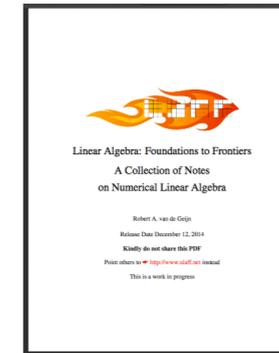
A constantly changing e-book (PDF) with programming exercises. For now, the document will give you an idea of what is coming. If you want to be kept informed of major milestones, join the [ULAFF-On](#) google group. Some of the programming activities are now available on github. Stand by!

Funded in part by the [National Science Foundation](#) (grant CCF-1714091).

Linear Algebra: Foundations to Frontiers - Notes on Numerical Linear Algebra

Notes for a graduate level numerical linear algebra course.

Download! [Linear Algebra: Foundations to Frontiers - Notes on Numerical Linear Algebra](#)



An e-book (PDF) that is a collection of notes written for an introductory graduate level course on Numerical Linear Algebra.

Funded in part by the [National Science Foundation](#) (grant ACI-1148125).

a stack for teaching high performance gemm

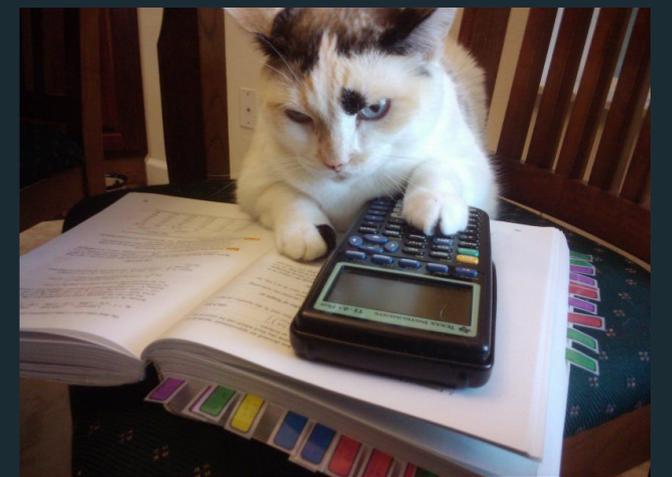
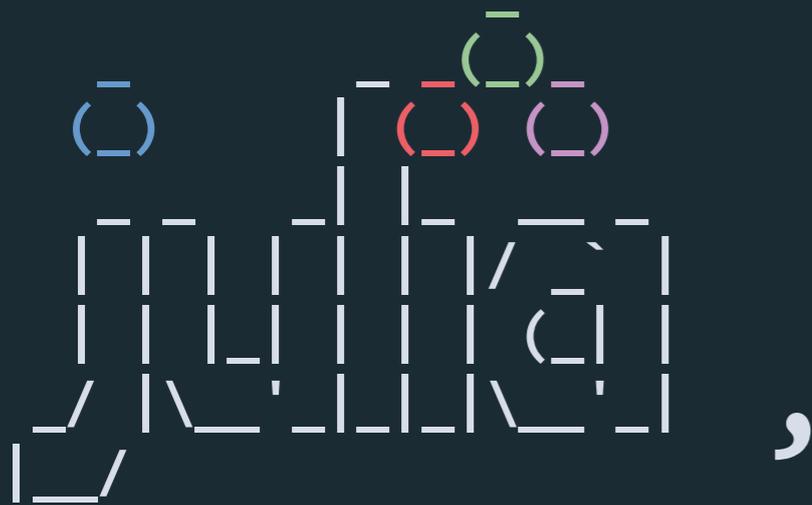
matlab

pseudocode
prototyping
exploration
testing
benching
plotting

c

down to the metal
transparent performance model





, &

