The Julia Programming Language – an introduction

Tony Kelman @tkelman

eMLDSC 3/02/16

Getting started

- Hosted in-browser notebook at https://www.juliabox.org/
- Or download an installer from http://julialang.org/downloads/
- For local install, open Julia REPL then run

```
Pkg.add("IJulia")
using IJulia
notebook()
```

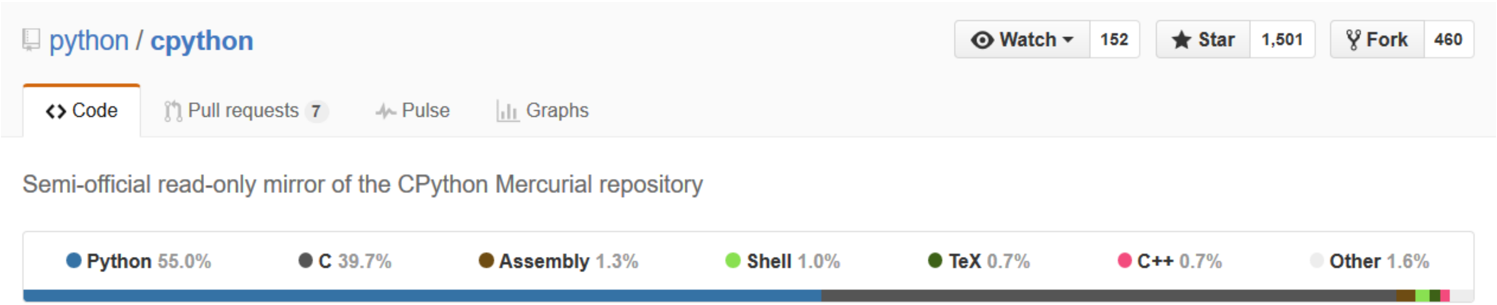
- If you already have Jupyter installed it should use that, this creates a new Julia kernel profile

Why a new language?

- Have a problem to solve and want a computer to do the work
- Has anyone else already solved the exact same problem and shared their code? Use it!
- Does that code meet your requirements?
  - Easy enough to use?
  - Good enough performance?
  - Can you understand the code if you find a bug?
  - Can it be extended to handle new data structures or modified algorithms?
- (if so, you're lucky, thank the author and help fix a bug or something)

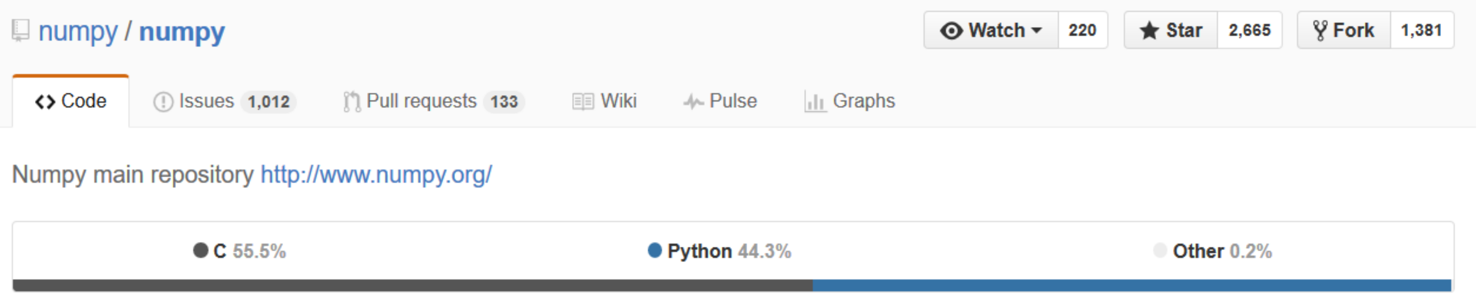
2-language problem

- Performance critical parts in C, user friendly wrappers in Python



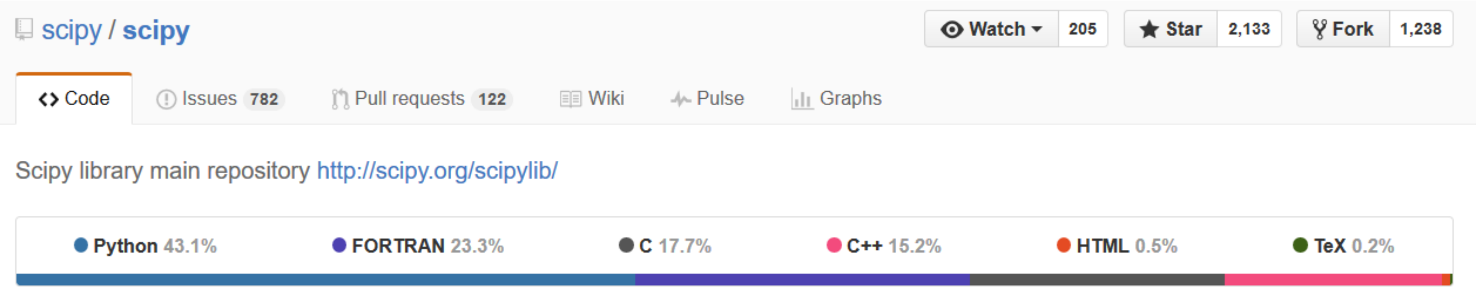
Semi-official read-only mirror of the CPython Mercurial repository

● Python 55.0% ● C 39.7% ● Assembly 1.3% ● Shell 1.0% ● TeX 0.7% ● C++ 0.7% ● Other 1.6%



Numpy main repository http://www.numpy.org/

● C 55.5% ● Python 44.3% ● Other 0.2%



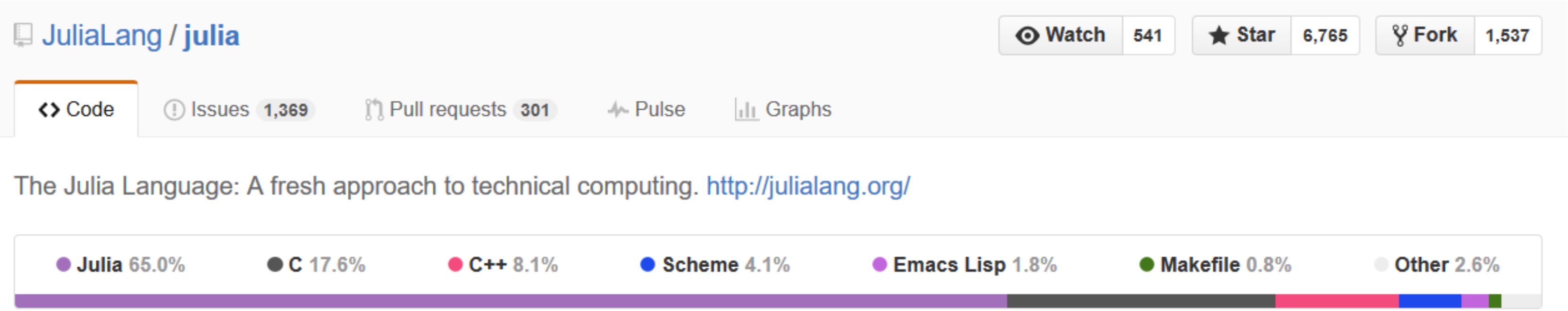
Scipy library main repository http://scipy.org/scipylib/

● Python 43.1% ● FORTRAN 23.3% ● C 17.7% ● C++ 15.2% ● HTML 0.5% ● TeX 0.2%

Can we do better?

julia

- Python, Numpy, Scipy, etc have been around a long time
- Many new technologies for making dynamic languages fast
- Just in time compilation to native code
- LLVM compiler framework, used by Clang etc
- Julia's goal: write like Python, run like C, easily access existing libraries and legacy code



JuliaLang / julia

The Julia Language: A fresh approach to technical computing. http://julialang.org/

● Julia 65.0% ● C 17.6% ● C++ 8.1% ● Scheme 4.1% ● Emacs Lisp 1.8% ● Makefile 0.8% ● Other 2.6%

Demo