

# Intro To Julia: A Great Language for Data Analytics

Steven Reynolds



slides at [www.slreynolds.net](http://www.slreynolds.net)

# Introduction

- Julia is a great\* general purpose language
- Julia is *fast*
- Julia is good at math
- Julia has powerful plotting packages
- DataFrames are good for statistics

\*Admittedly subjective

# Why Language speed matters

- Switching or combining languages is *painful and a source of bugs*
- Application performance matters
  - Many problems have large datasets
  - End users won't use slow systems
- *Want to use same language for both high level glue code as low level loops*

- Most programming languages fail at math.
- Difficult to make primitive operations both fast *and* flexible.
- Other languages choose either fast or flexible.
- Either choice is a failing proposition.

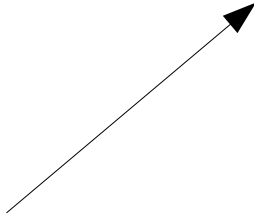
C	Fast and inflexible
Java	Fast and inflexible
Scala	Slow and mostly inflexible
Matlab	Extremely slow
JavaScript	Fast-ish and <b>very</b> inflexible

Julia is both fast\* and flexible

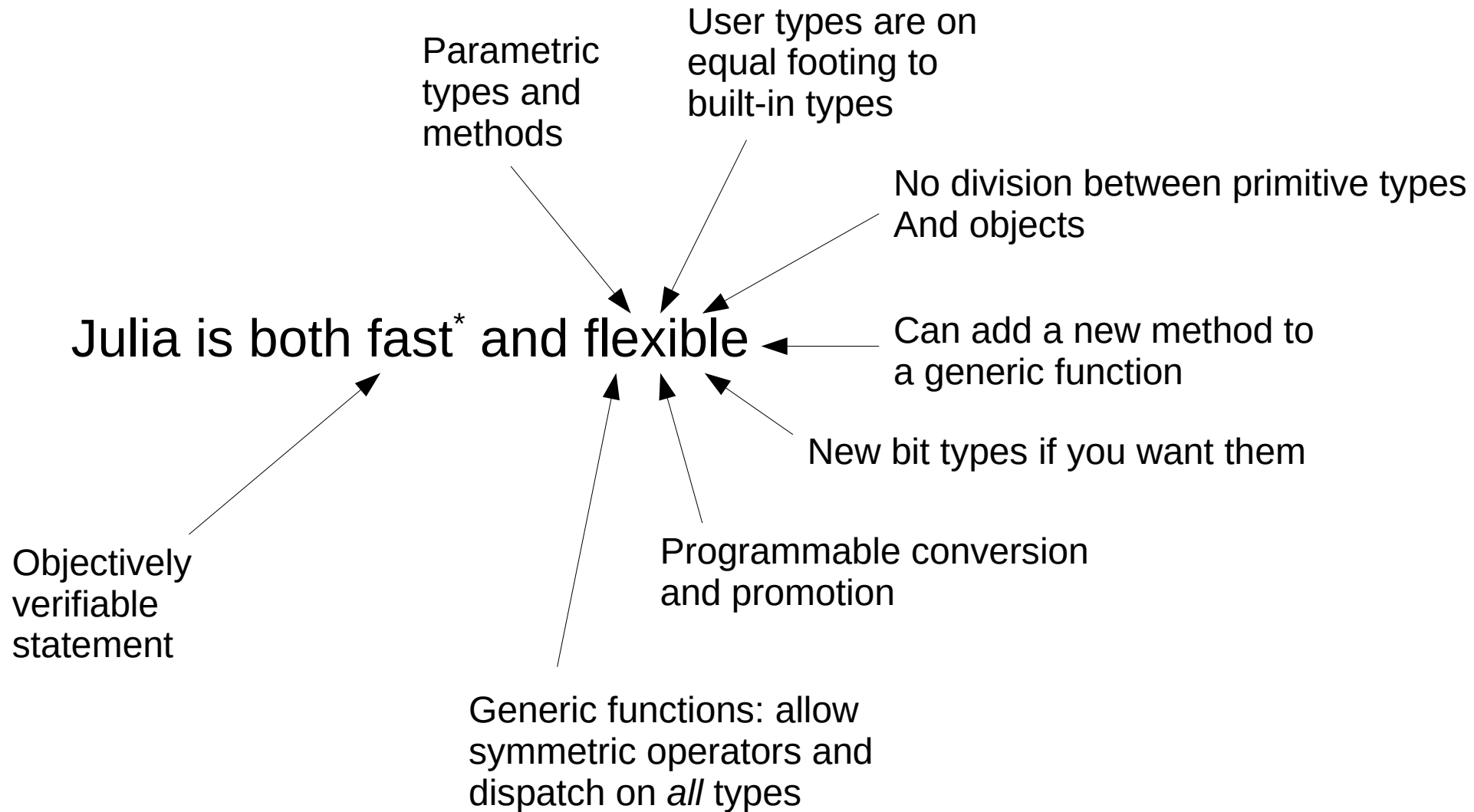
\*Julia does do JIT compilation and Garbage collection

Julia is both fast\* and flexible

Objectively  
verifiable  
statement



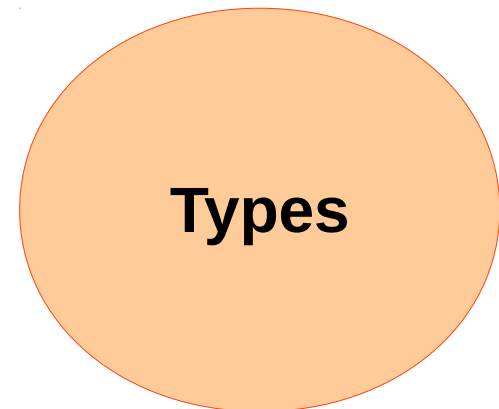
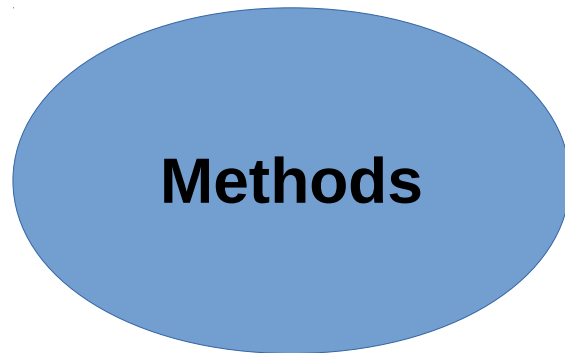
\*Julia does do JIT compilation and Garbage collection



\* Julia does do JIT compilation and Garbage collection

# Julia is ...

- Julia is Not Object Oriented
- Julia uses generic functions
- Julia has a type hierarchy
- Methods are *not* attached to types





## Types

## Methods

	Cat	Tiger	Weasel
Speak			
eat			
run			

# Object Oriented

	Cat	Tiger	Weasel
Speak			
Eat			
Run			

# Generic Functions

	Cat	Tiger	Weasel
Speak			
Eat			
Run			

# Math

- Generic functions are good for math

$$1.0 + 3$$

- In an Object Oriented Language, which argument owns the +?
- Having one argument own + is a problem.
  - + is supposed to commute.
  - The OO paradigm fights against correct code.
- With generic functions, none of the arguments own +, it's a first class entity
- Julia dispatches on *all* the types of the method arguments

# Math

- Julia gets other math things very right
- Has complex, vector and matrix types built-in
- Expressions: promotion and conversion are programmable by anyone
  - When you add types they can seamlessly integrate into expressions

# Plotting Packages

Winston	Simplest, robust and fast. Built ontop of cairo graphics. Might be painful on Windows.
PyPlot	Full featured. Uses Python and MatLibPlot.
Gadfly	Newest, sophisticated. Active. To get an on-screen plot, uses browser based atom library.

# DataFrames

- Like tables for in memory data
- Like pandas
- Common for data analysis, statistical problems
- Can do selection and projection
- Can apply closures

# Parallelism

- Julia has very interesting capabilities for distributed computing and parallelism built-in
  - Based on message passing, but doesn't look like it
  - One sided
  - Remote calls and remote references

```
nheads = @parallel (+) for i=1:200000000
    int(rand(Bool))
end
```



# Environments

Julia Studio	IDE	Abandoned – not recommended.
Juno	IDE	Built ontop of lighttable – active IDE. Easy to install.
IJulia	Browser based	Built ontop of IPython & Jupyter. Installation is involved. Plots work nicely.
Built-in repl	repl	The built-in Julia REPL is nice.
Sublime	editor	Front end using sublime editor with IJulia backend. Recommend just using it as an editor – REPL interaction not completely functional.

# Julia Speed

- Julia uses llvm to compile code down to native binary
- Compile happens “just in time” – first time method is encountered
- Compiler is given all known information about argument types. Clever type inferencing.
- Compile output is specialized for known types

# Hints

- [julialang.org](http://julialang.org)
- Start with [julialang.org](http://julialang.org) > Docs > Getting Started
- Learn Julia in Y Minutes & The Julia Express
- Examples in Julia source <https://github.com/JuliaLang/julia>

Be sure to get the version of source  
that matches your julia binary

- Better examples: Forio Julia tutorials (don't use Forio Julia Studio)
- Many 3<sup>rd</sup> party packages docs are hosted on [readthedocs.org](http://readthedocs.org)

# Future


- Expect some changes going from 0.3 to 0.4
  - NA to Nullable
  - Changes (redesign) to DataFrame
  - String and Dictionary may change names
  - None and Nothing will change names
  - Precompiled modules

Parallel Computing — Julia Language 0.4.0-pre documentation - Mozilla Firefox

Parallel Computing ...

Search

Most Visited Getting Started



Search docs

- Introduction
- Getting Started
- Variables
- Integers and Floating-Point Numbers
- Mathematical Operations and Elementary Functions
- Complex and Rational Numbers
- Strings

Docs » Parallel Computing [Edit on GitHub](#)

## Parallel Computing

Most modern computers possess more than one CPU, and several computers can be combined together in a cluster. Harnessing the power of these multiple CPUs allows many computations to be completed more quickly. There are two major factors that influence performance: the speed of the CPUs themselves, and the speed of their access to memory. In a cluster, it's fairly obvious that a given CPU will have fastest access to the RAM within the same computer (node). Perhaps more surprisingly, similar issues are relevant on a typical multicore laptop, due to differences in the speed of main memory and the cache. Consequently, a good multiprocessing environment should allow

v: latest

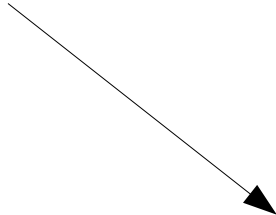
on a

the

que

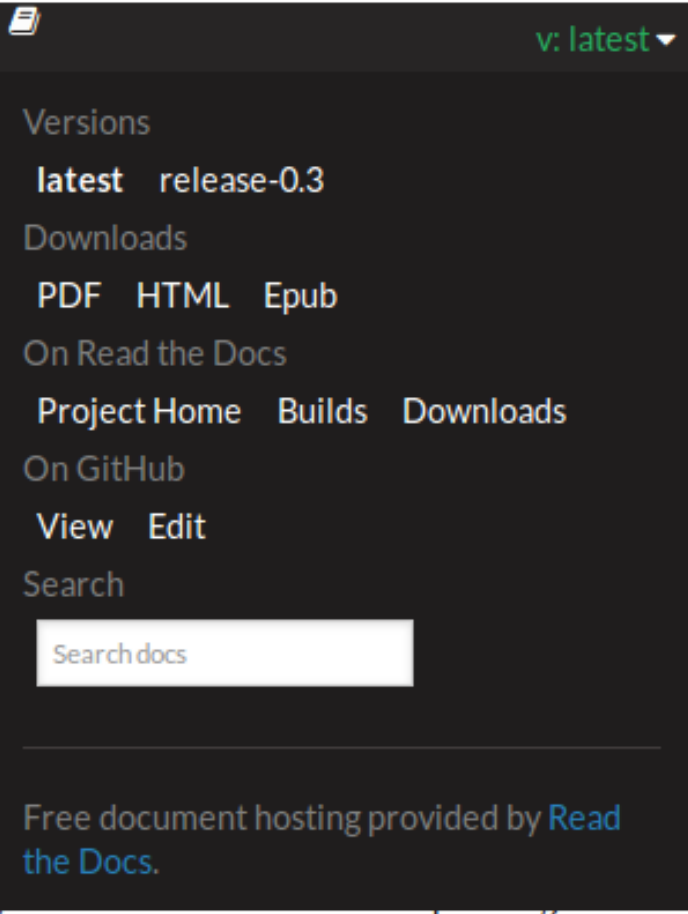


AW



ting

[Edit on GitHub](#)



v: latest ▾

Versions

**latest** release-0.3

Downloads

PDF HTML Epub

On Read the Docs

Project Home Builds Downloads

On GitHub

View Edit

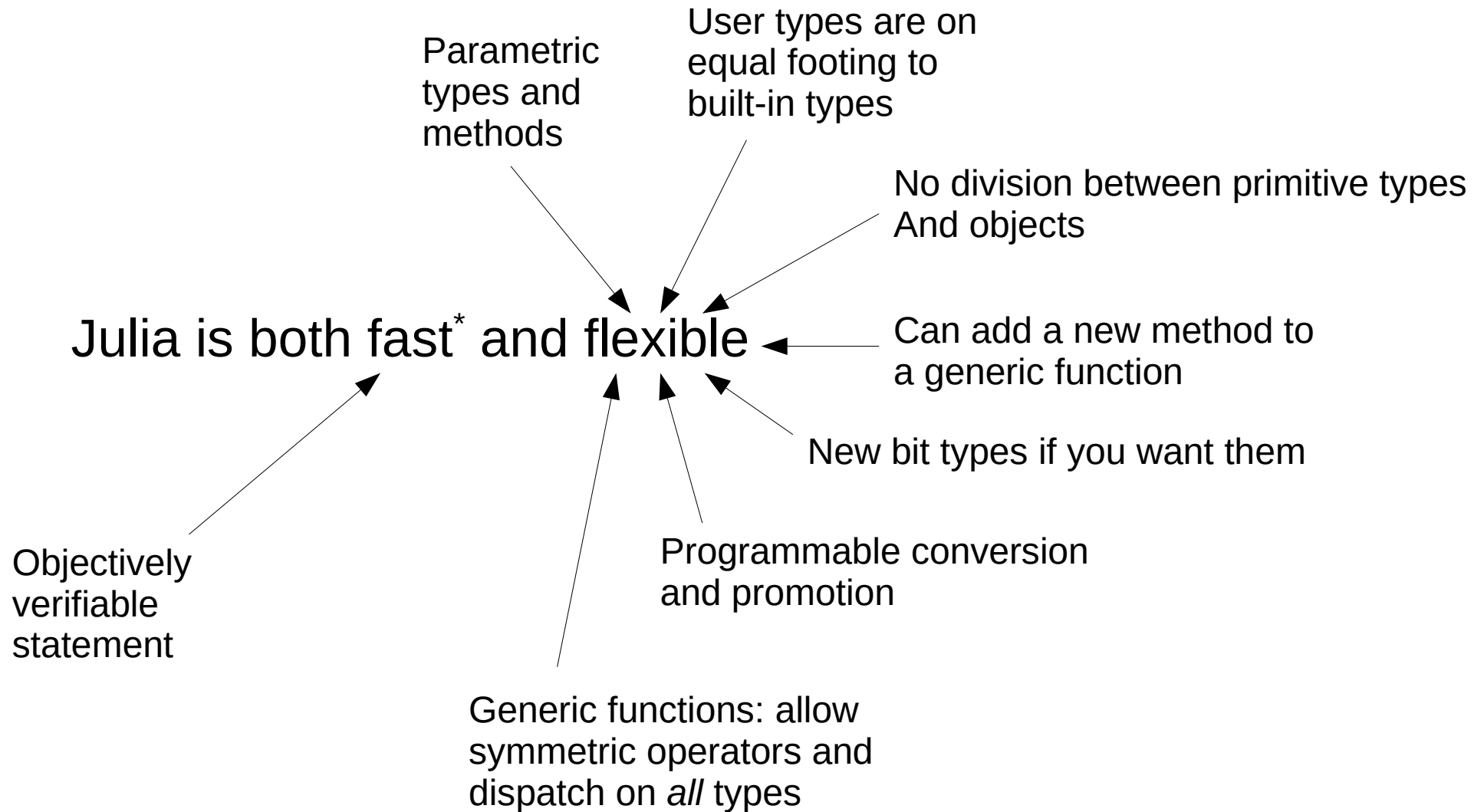
Search

---

Free document hosting provided by [Read the Docs](#).

# Conclusion

- Julia is a great general purpose language
  - It is not Object Oriented
  - Uses generic functions
- Julia is *fast*
- Julia is good at Math
- Julia has nice plotting packages
- DataFrames are good for statistics



\* Julia does do JIT compilation and Garbage collection