Clojure
It's the new Ruby

Micah Martin
@slagyr
8th Light, Inc.

Intro

Ruby

- Object Oriented (near pure)
- interpreted (by C code)
- developer friendly
- JRuby (runs on JVM)

Clojure

- Dialect of Lisp
- Functional
- Runs on JVM

Why is Clojure the new Ruby?

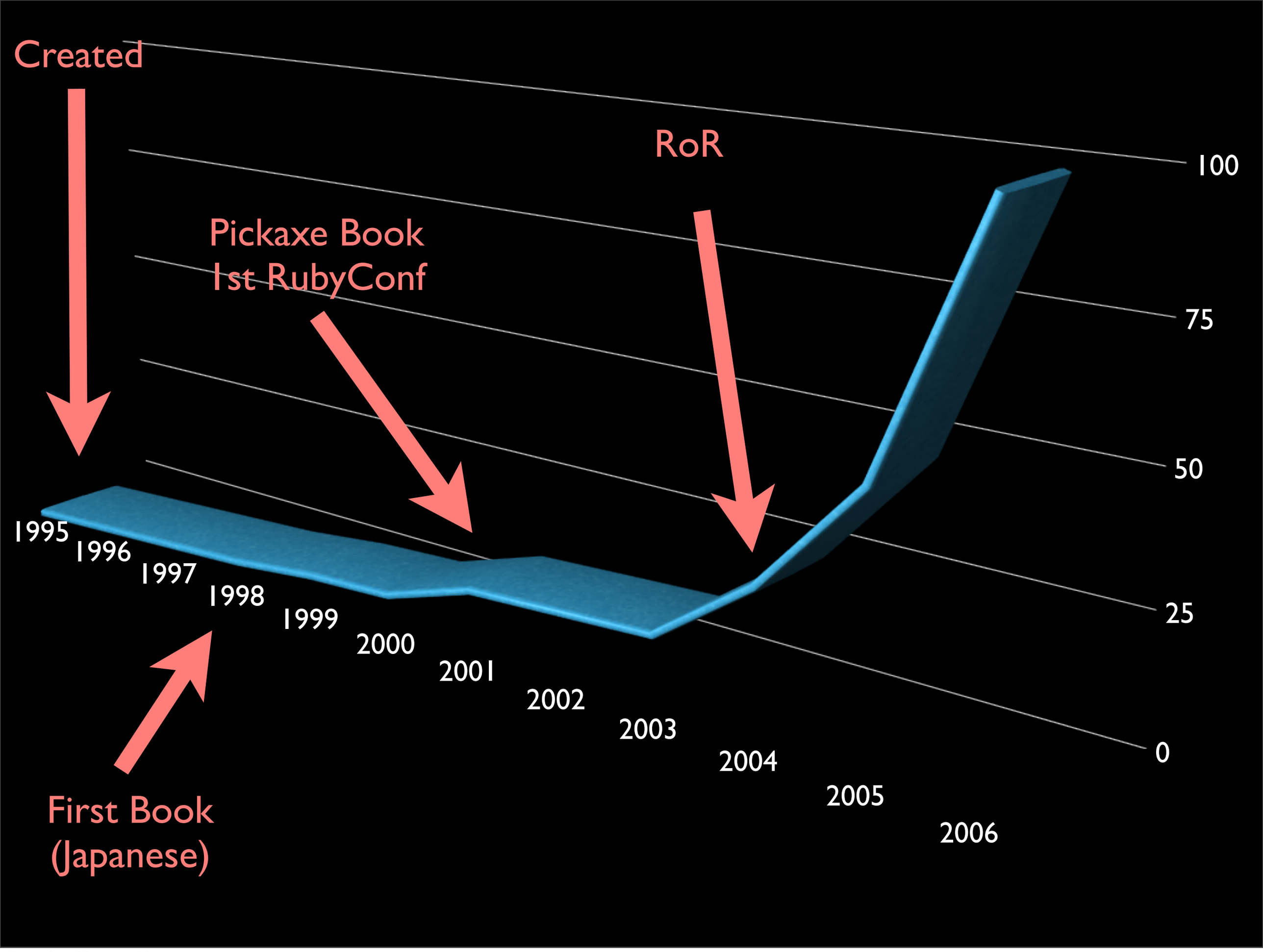
Mode

vogue / fashion

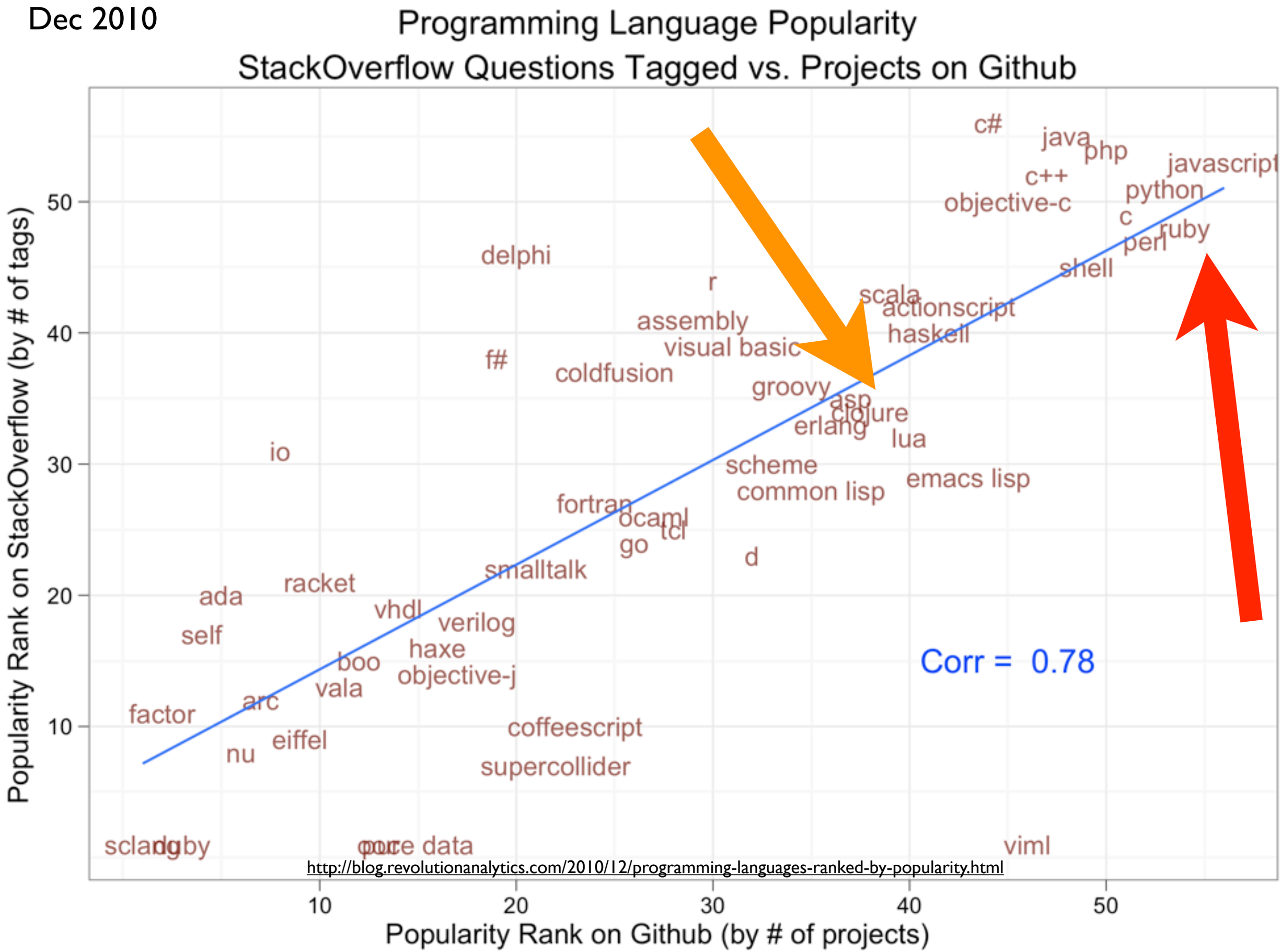
Try Blue

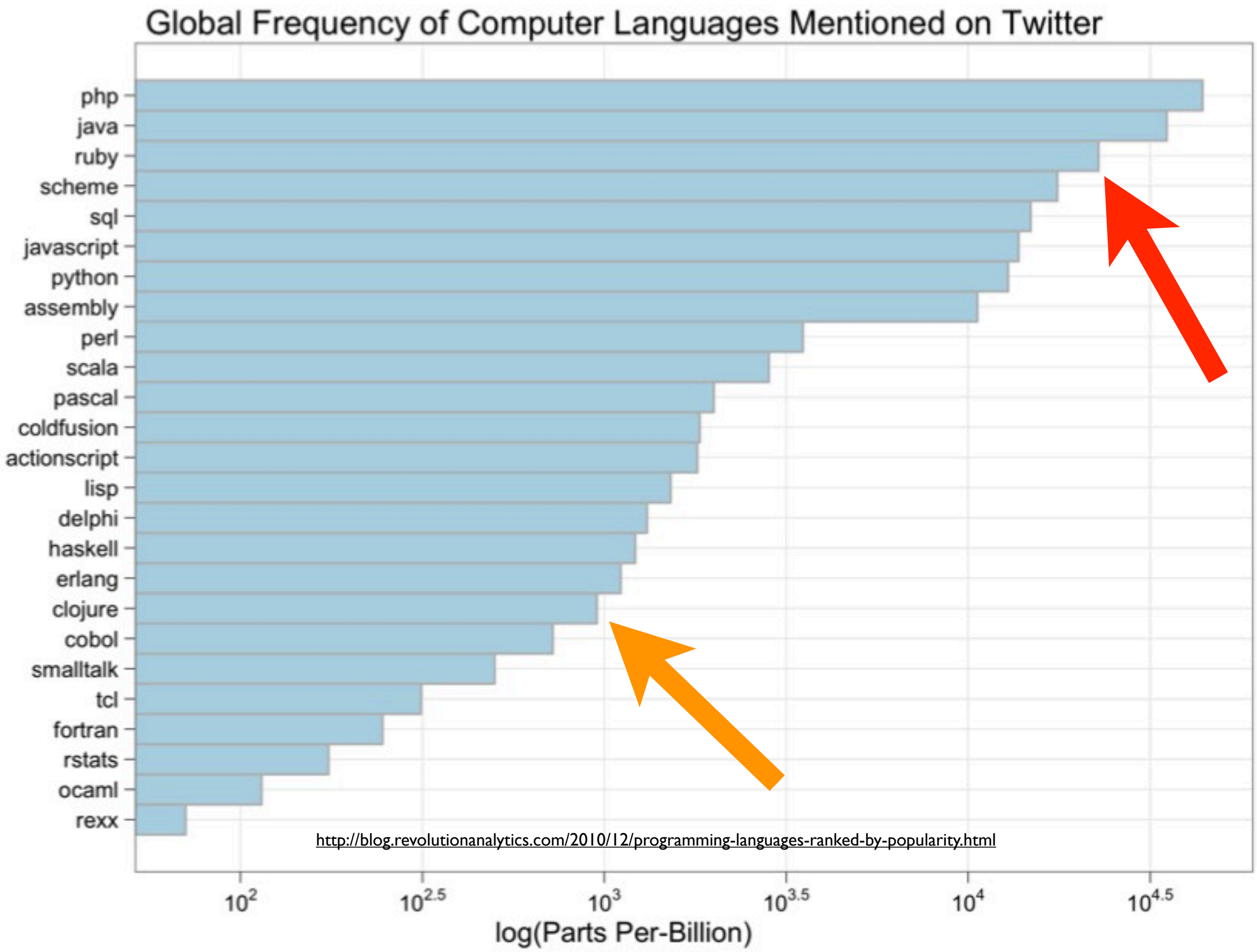
Disney - Wall-E motion picture

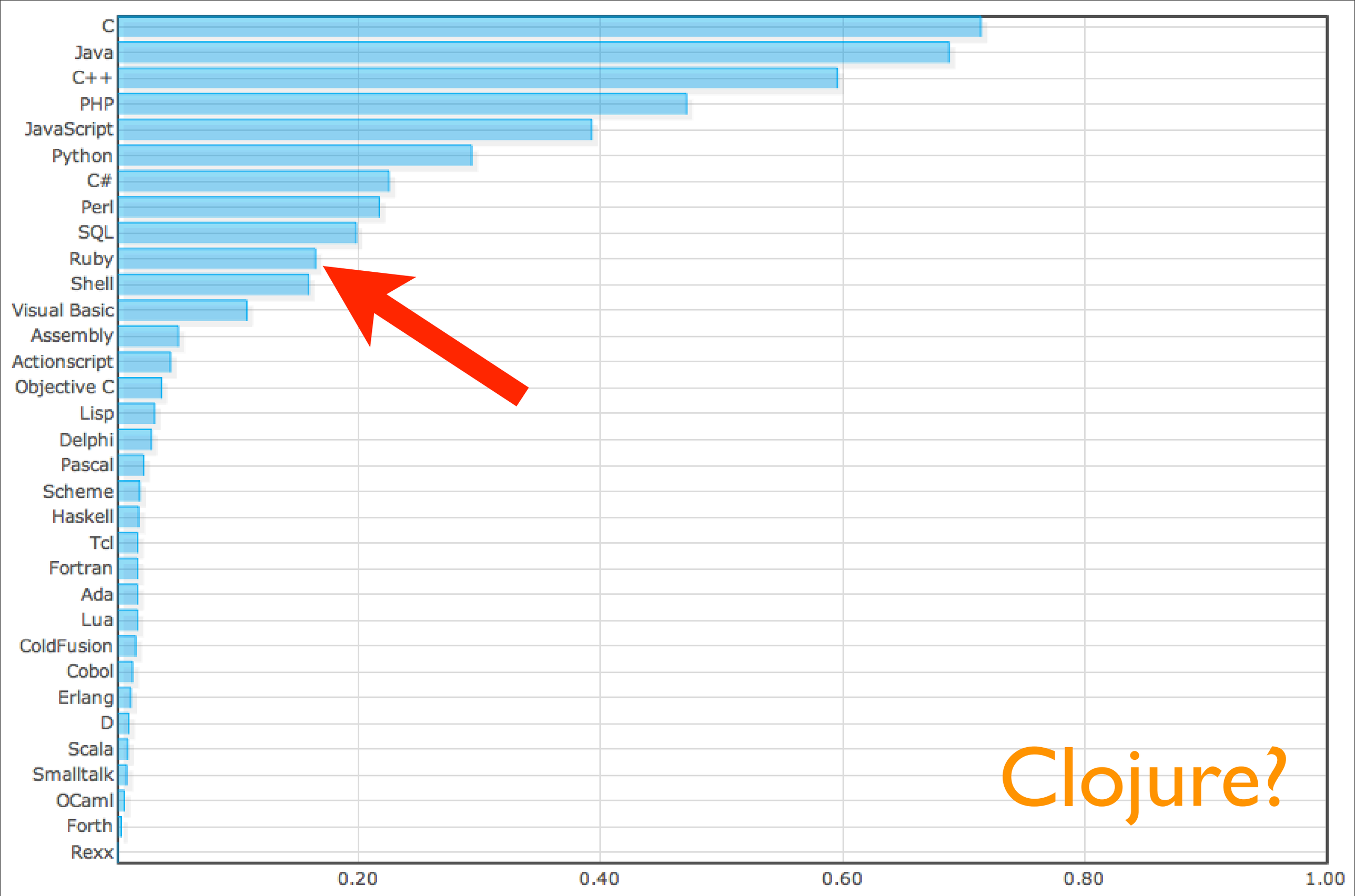
It's the new Red!


Created
RoR
Pickaxe Book
1st RubyConf
First Book
(Japanese)
100
75
50
25
0
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006

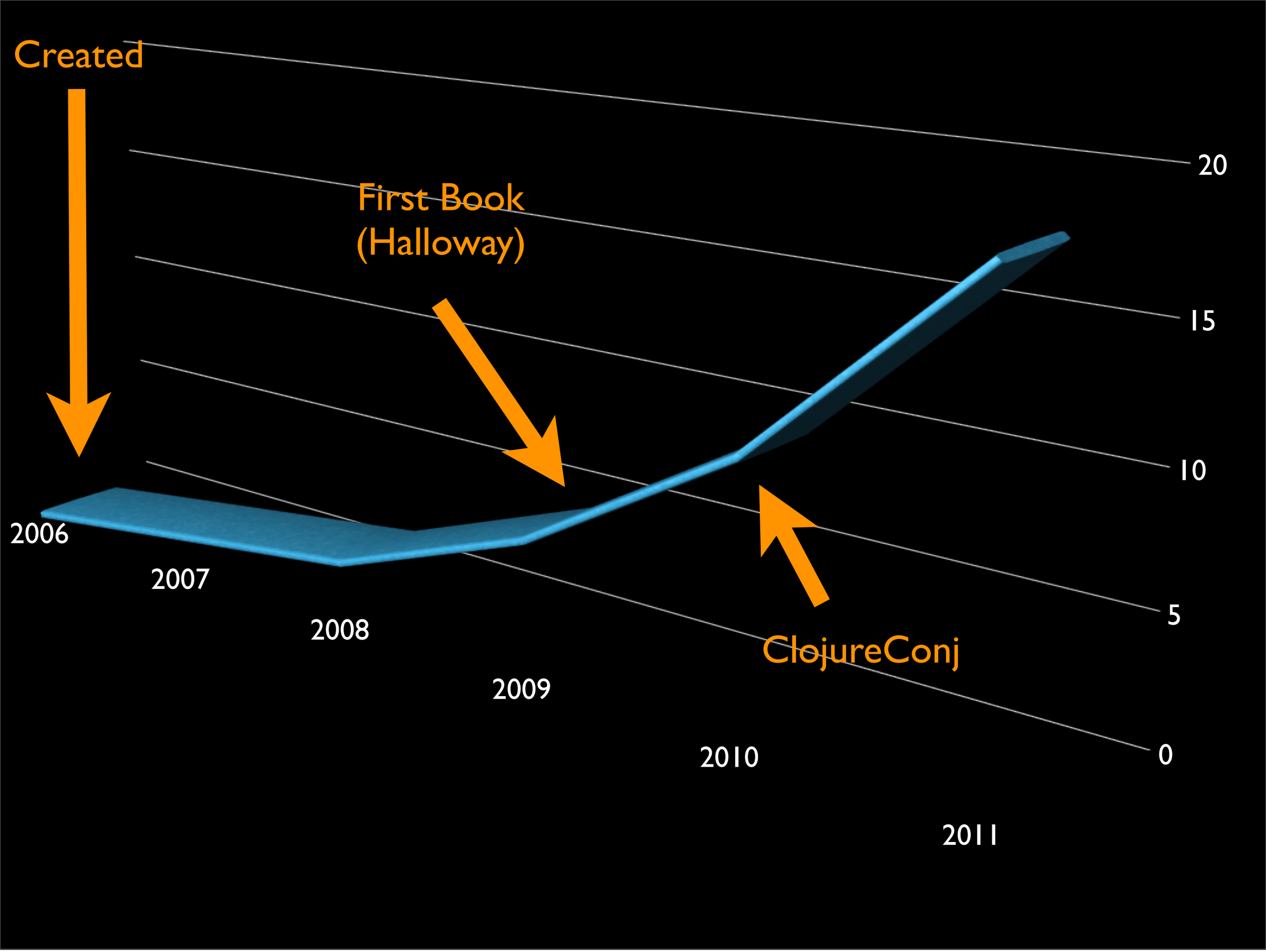
Dec 2010


Programming Language Popularity
StackOverflow Questions Tagged vs. Projects on Github
Popularity Rank on StackOverflow (by # of tags)
Popularity Rank on Github (by # of projects)
10
20
30
40
50
c#
java
php
javascript
c++
python
objective-c
c
ruby
perl
shell
delphi
r
scala
actionscript
assembly
haskell
visual basic
f#
coldfusion
groovy
asp
clojure
erlang
lua
io
scheme
common lisp
emacs lisp
fortran
ocaml
tcl
go
d
smalltalk
racket
ada
vhdl
verilog
self
haxe
boo
objective-j
vala
arc
factor
coffeescript
eiffel
nu
supercollider
viml
Corr = 0.78
http://blog.revolutionanalytics.com/2010/12/programming-languages-ranked-by-popularity.html

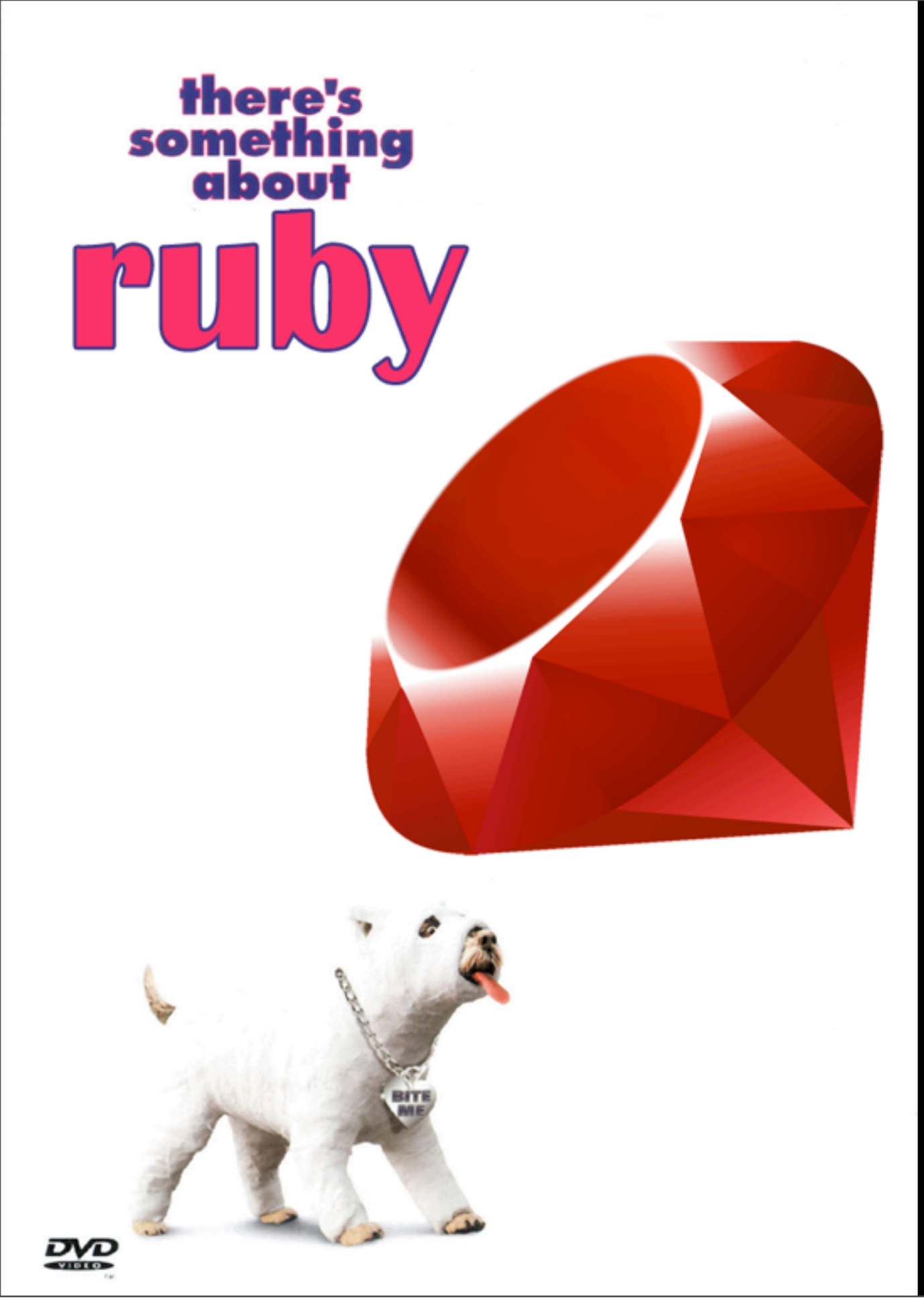

Global Frequency of Computer Languages Mentioned on Twitter
php
java
ruby
scheme
sql
javascript
python
assembly
perl
scala
pascal
coldfusion
actionscript
lisp
delphi
haskell
erlang
clojure
cobol
smalltalk
tcl
fortran
rstats
ocaml
rexx
http://blog.revolutionanalytics.com/2010/12/programming-languages-ranked-by-popularity.html
10^2
$10^{2.5}$
10^3
$10^{3.5}$
10^4
$10^{4.5}$
log(Parts Per-Billion)



http://www.langpop.com


Created
First Book
(Halloway)
ClojureConj
20
15
10
5
0
2006
2007
2008
2009
2010
2011

Language Features



What?

- Interpreted
- Expressive Syntax
- Monkey Patching
- Dynamic Typing
- Meta Programming
- Pure OO

How does Clojure Measure up?

- ✓ Interpreted
- ✓ Expressive Syntax
- **ALMOST** Monkey Patching
- ✓ Dynamic Typing
- ✓ Meta Programming
- **ALMOST** Pure OO

Prime Factors Code

Ruby #

```ruby
module PrimeFactors

  def self.of(n)
    factors = []
    divisor = 2
    while n > 1
      while n % divisor == 0
        factors << divisor
        n /= divisor
      end
      divisor += 1
    end
    return factors
  end

end
```

Clojure #

```clojure
(ns prime-factors)

(defn factors-of [n]
  (loop [factors [] divisor 2 n n]
    (if (<= n 1)
      factors
      (if (= 0 (rem n divisor))
        (recur
          (conj factors divisor)
          divisor
          (/ n divisor))
        (recur
          factors
          (inc divisor)
          n)))))
```

Monkey Patching in Clojure? (option #1)

Clojure #

```clojure
(defn foo []
  (println "FOO!"))

(binding [foo #(println "BAR!")]
  (foo))
```

output: BAR!

Monkey Patching in Clojure? (option #2)

Clojure #

```clojure
(defprotocol Animal
  (speak [_]))

(extend-type String
  Animal
  (speak [_] "Woof!"))

(speak "Some String")
```

result: “Woof!”

Meta-Programming in Clojure?

Clojure is Object Oriented?

- Yes

Bonus Features

Homoiconicity

... is a property of some programming languages, in which the primary representation of programs is also a data structure of the language itself.

Clojure #

```
(1 2 3)

(list 1 2 3)
```

Homoiconicity lends to...

- Easily generating Clojure code
  - Macros!
- Easily reading Clojure code
- Easily manipulating Clojure code

Macro Example

Clojure #

```
(defmacro cond [& clauses]
    (when clauses
      (list 'if (first clauses)
            (if (next clauses)
                (second clauses)
                (throw (IllegalArgumentException.
                        "cond requires an even number of forms")))
            (cons 'clojure.core/cond (next (next clauses))))))
```

... it generates nested if calls

STM (Software Transactional Memory) and Mutation Management

atom

simple, singular mutations

Clojure #

```clojure
(def a (atom 0))

(swap! a inc)

@a
```

result: 1

agent

asynchronous, serialized mutations

Clojure #

```clojure
(def a (agent 0))

(send a inc)
(send a inc)

@a
```

result: Don't know! Either 0, 1, or 2

ref

transactional, synchronized mutations

Clojure #

```clojure
(def a (ref 1))
(def b (ref 10))

(dosync
  (alter a inc)
  (alter b inc))

(+ @a @b)
```

result: 13

New Paradigm

Functional Programming

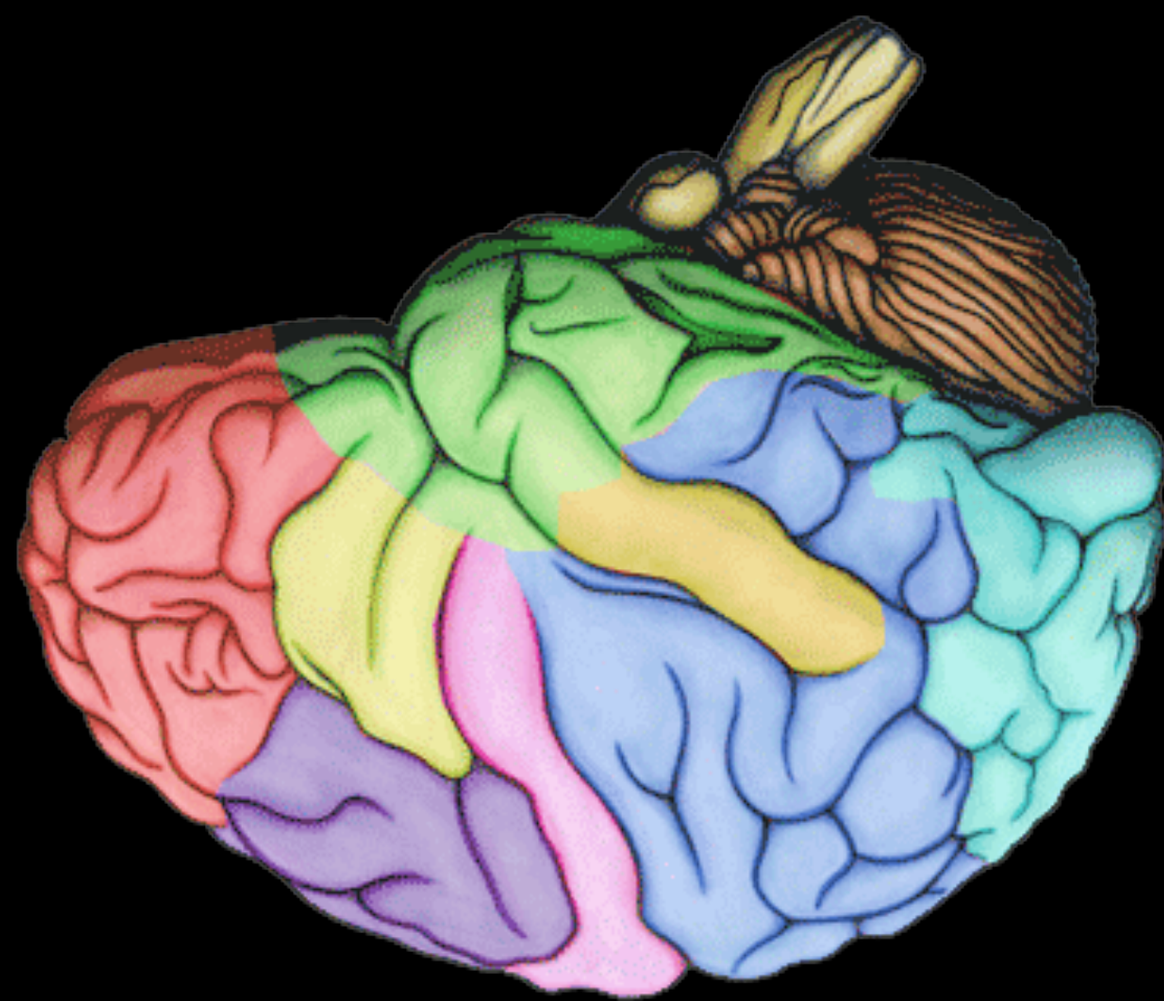
Functions are First Class citizens

first class image by United

No mutable state

wolverine by marvel comics

New Paradigm...



Different and Fun!

http://northernrockiesneurosurgeons.com/brain.htm

The Last Programming Language

http://www.cleancoders.com/codecast/theLastProgrammingLanguage/show

Ruby on

What?

gaeshi

Thanks!
Questions?

Micah Martin
@slagyr
8th Light, Inc.

https://github.com/slagyr/gaeshi