

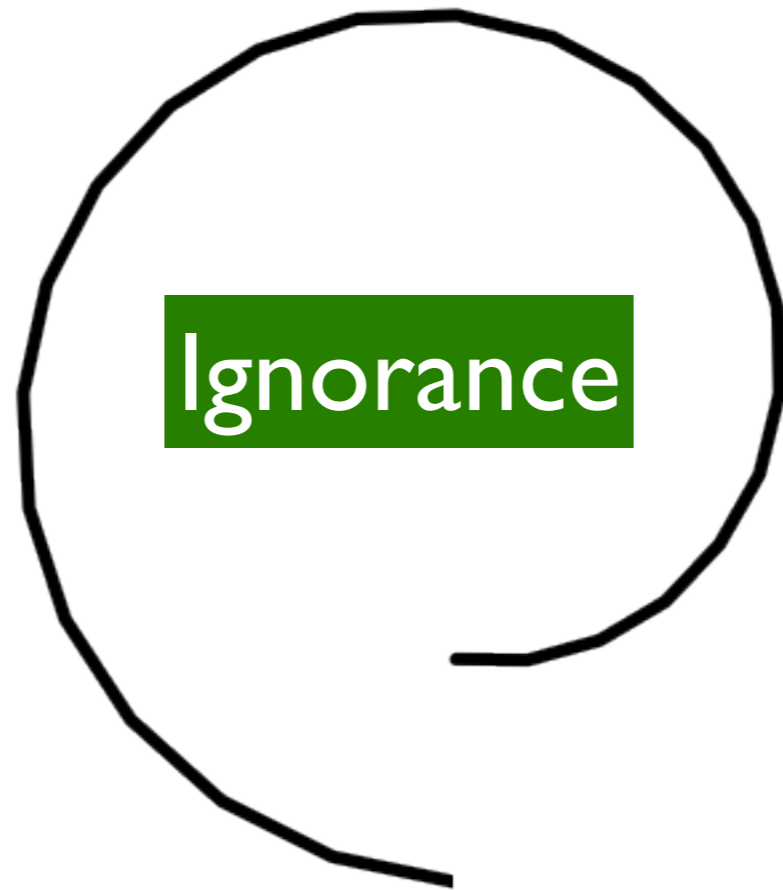
The Prejudgement of Programming Languages

@garybernhardt

10 Years of Failure and Bad Ideas

@garybernhardt

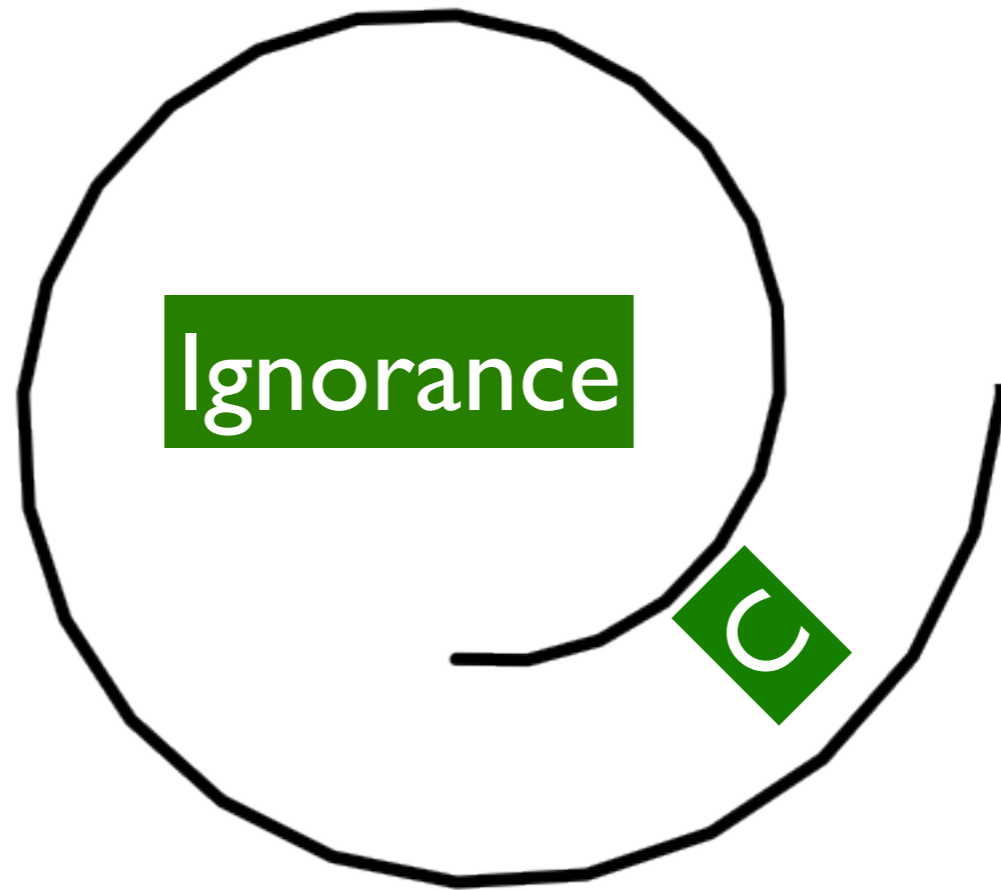
Pre-2001



2001

C is the best thing

Important: Java sucks



2003

```
; The Lisp defined in McCarthy's 1960 paper, translated into CL.
; Assumes only quote, atom, eq, cons, car, cdr, cond.
; Bug reports to lispcode@paulgraham.com.
```

```
(defun null. (x)
  (eq x '()))

(defun and. (x y)
  (cond (x (cond (y 't) ('t '()))))
        ('t '())))

(defun not. (x)
  (cond (x '())
        ('t 't)))

(defun append. (x y)
  (cond ((null. x) y)
        ('t (cons (car x) (append. (cdr x) y)))))

(defun list. (x y)
  (cons x (cons y '())))

(defun pair. (x y)
  (cond ((and. (null. x) (null. y)) '())
        ((and. (not. (atom x)) (not. (atom y)))
         (cons (list. (car x) (car y))
               (pair. (cdr x) (cdr y)))))

(defun assoc. (x y)
  (cond ((eq (caar y) x) (cadar y))
        ('t (assoc. x (cdr y)))))

(defun eval. (e a)
```

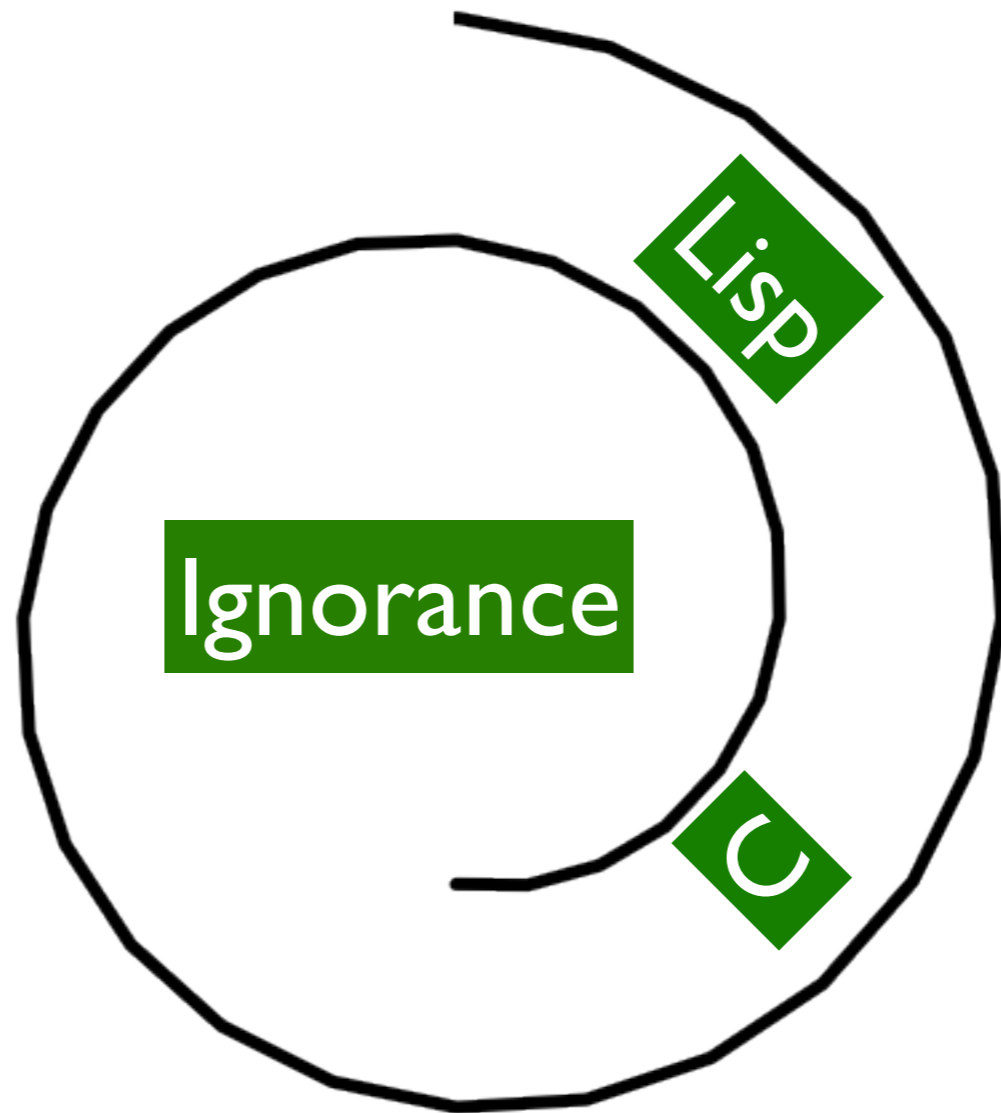
```
(cond
  ((atom e) (assoc. e a))
  ((atom (car e))
   (cond
    ((eq (car e) 'quote) (cadr e))
    ((eq (car e) 'atom) (atom (eval. (cadr e) a)))
    ((eq (car e) 'eq) (eq (eval. (cadr e) a)
                          (eval. (caddr e) a)))
    ((eq (car e) 'car) (car (eval. (cadr e) a)))
    ((eq (car e) 'cdr) (cdr (eval. (cadr e) a)))
    ((eq (car e) 'cons) (cons (eval. (cadr e) a)
                              (eval. (caddr e) a)))
    ((eq (car e) 'cond) (evcon. (cdr e) a))
    ('t (eval. (cons (assoc. (car e) a)
                    (cdr e))
           a))))
  ((eq (caar e) 'label)
   (eval. (cons (caddr e) (cdr e))
          (cons (list. (cadar e) (car e)) a)))
  ((eq (caar e) 'lambda)
   (eval. (caddr e)
          (append. (pair. (cadar e) (evlis. (cdr e) a))
                   a))))))

(defun evcon. (c a)
  (cond ((eval. (caar c) a)
         (eval. (cadar c) a))
        ('t (evcon. (cdr c) a))))

(defun evlis. (m a)
  (cond ((null. m) '())
        ('t (cons (eval. (car m) a)
                   (evlis. (cdr m) a)))))
```

"Crazy" language

Python



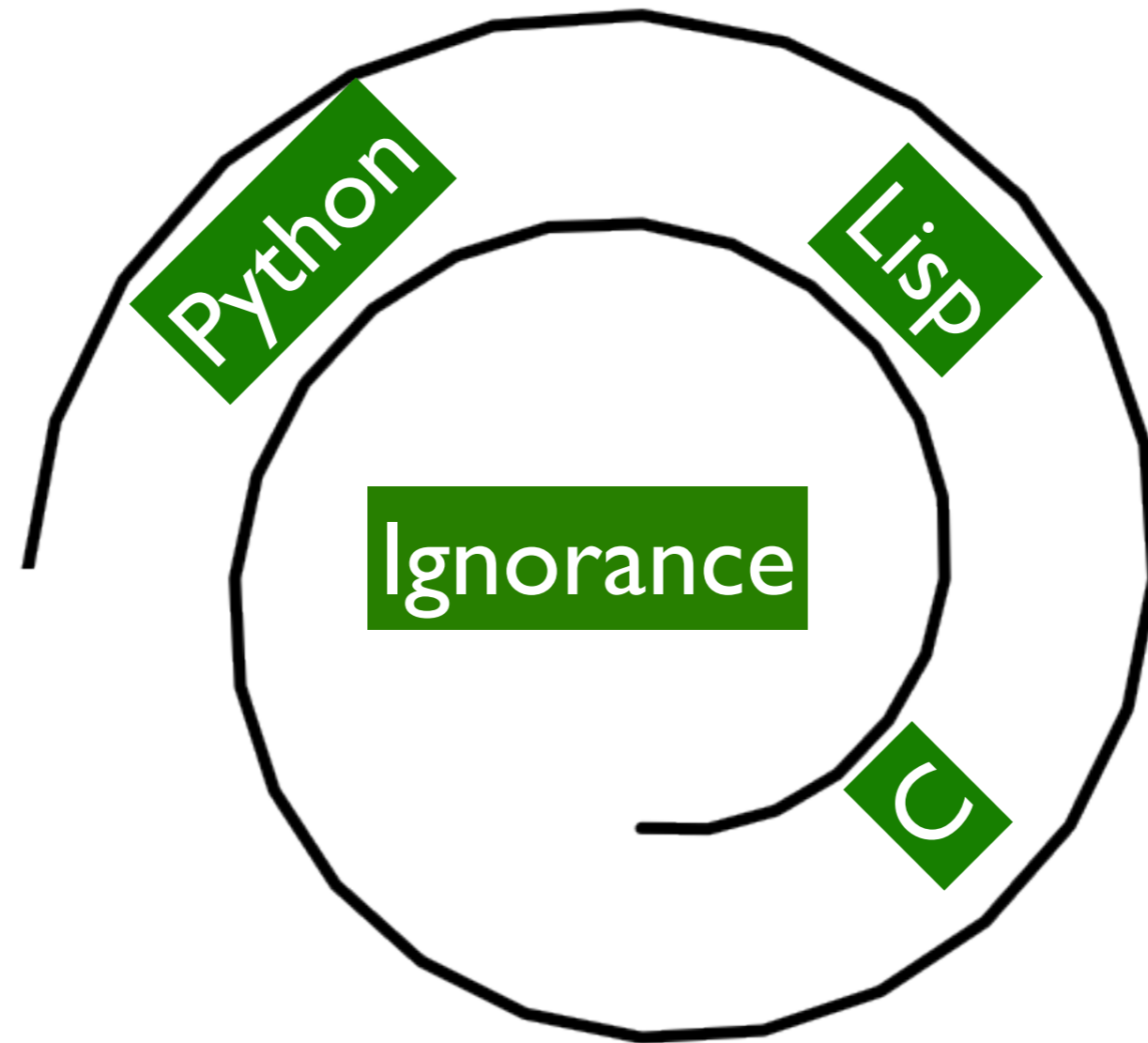
2006

BitBacker

Three years, 98.7% Python

Python <3

Exhausted



See:

"A Brief History of BitBacker"

"My Startup Talk From NWVDP 2010"

"Backup Is Hard; Let's Go Shopping"

2009

Ruby and Python, 50/50

"I can integrate a Python lib in ten minutes, a Ruby lib in an hour, and a JavaScript lib in a day, but it'll mysteriously break tomorrow."

"Note to Ruby guys: not all problems should be solved with cute syntax tricks. However, note to everyone else: some problems should."

"I thought up a new spec runner for Python, then realized that it was just an elaborate scheme for working around Python's lack of blocks."

Kernel (Module)

In: ./object.c
./lib/yaml.rb

ryan: You know how `Kernel.p` is a really convenient way to dump ruby

```
structures? The only downside is that it's not as legible as
YAML.
```

```
_why: (listening)
```



i am deeply touched
by this strange book,
Why's Poignant
Guide to Ruby.

seriously.
i feel great
inside.

like the cavity
of my body is
filled with an
avocado and
mayonnaise whip.

Kernel (Module)

In: `./object.c`
`./lib/yaml.rb`

ryan: You know how `Kernel.p` is a really convenient way to dump ruby

```
structures? The only downside is that it's not as legible as
YAML.
```

`_why`: (listening)

ryan: I know you don't want to urinate all over your users' namespaces.

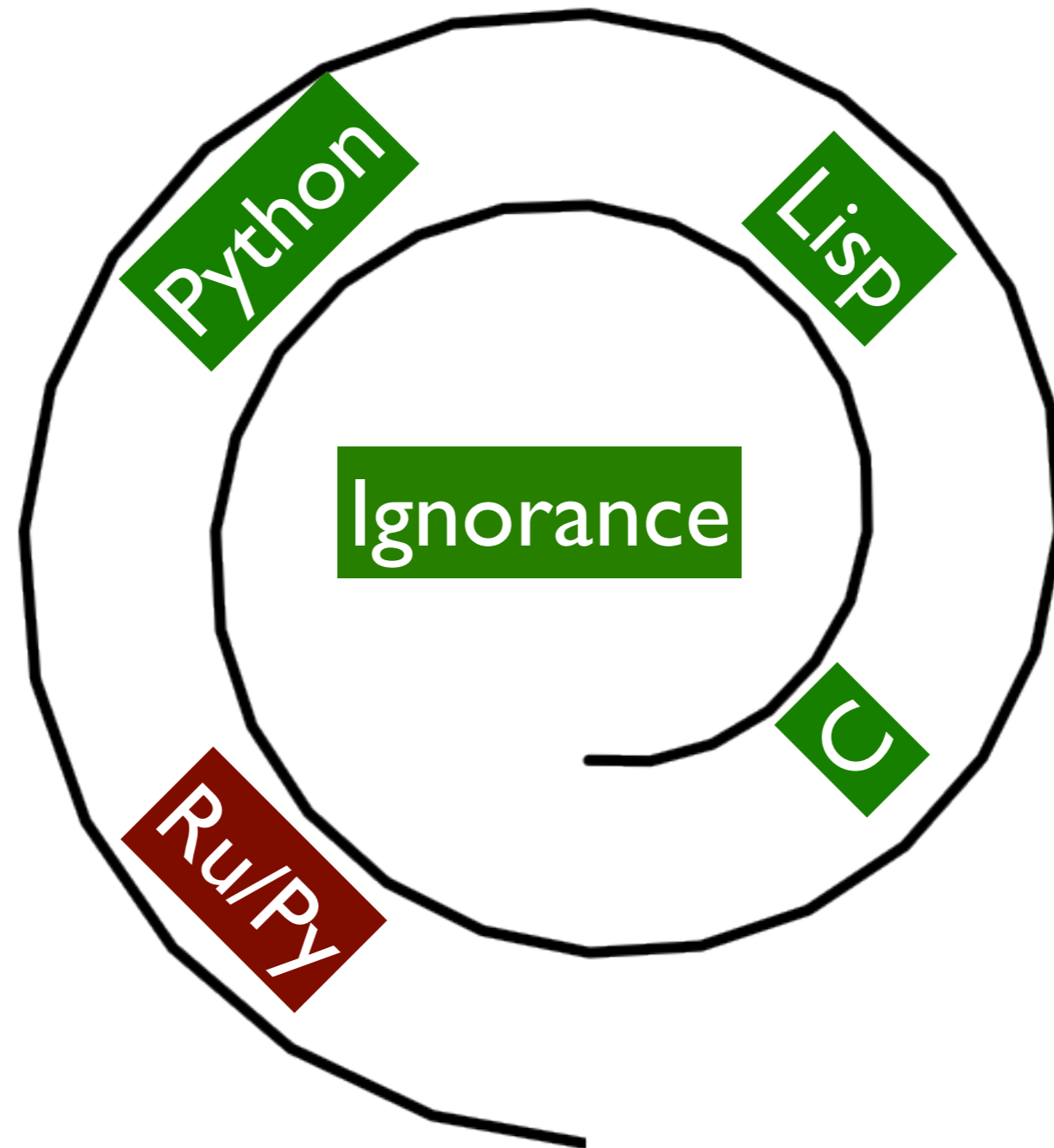
```
But, on the other hand, convenience of dumping for debugging is,
IMO, a big YAML use case.
```

`_why`: Go nuts! Have a pony parade!

ryan: Either way, I certainly will have a pony parade.

"This is your periodic reminder that Ruby's Kernel module docs begin with a misformatted discussion of urine and ponies."

"Ruby isn't serious"



Q2 2010


```
class TestCounter:
    def test_increment(self):
        c = Counter()
        old_count = c.count
        c.increment
        assertEquals(c.count, old_count + 1)
```

(Demo)

```
describe "Counter#increment"  
  it "increments by one" do  
    c = Counter.new  
    expect { c.increment }.to change { c.count }.by(1)  
  end  
end
```

Claim: "RSpec is confusing"

Feigned ignorance

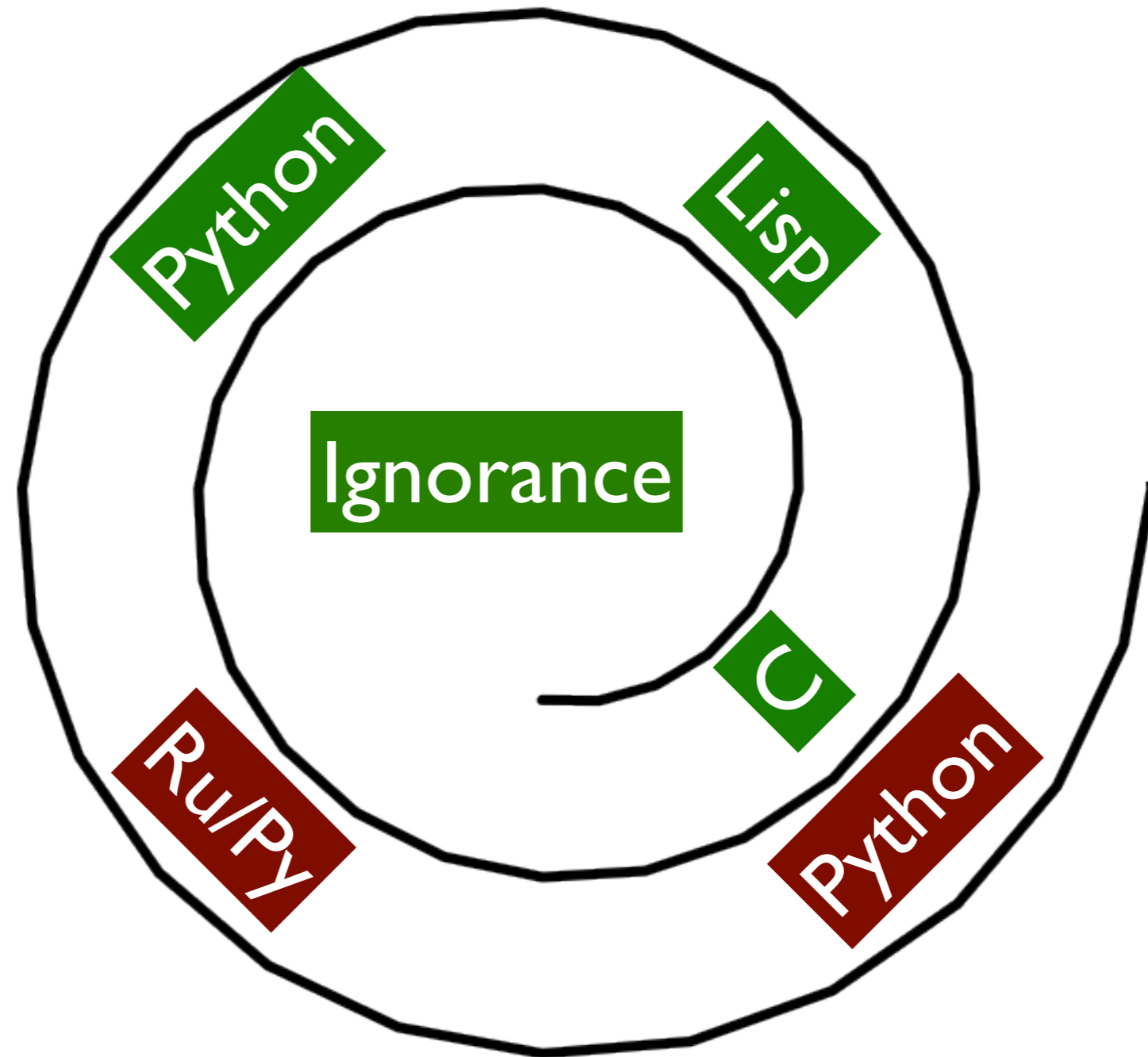
RSpec has won in Ruby

SUnit: 1994

```
def contact(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)
    else:
        form = ContactForm()
    return render_to_response('contact.html',
                              {'form': form})
```

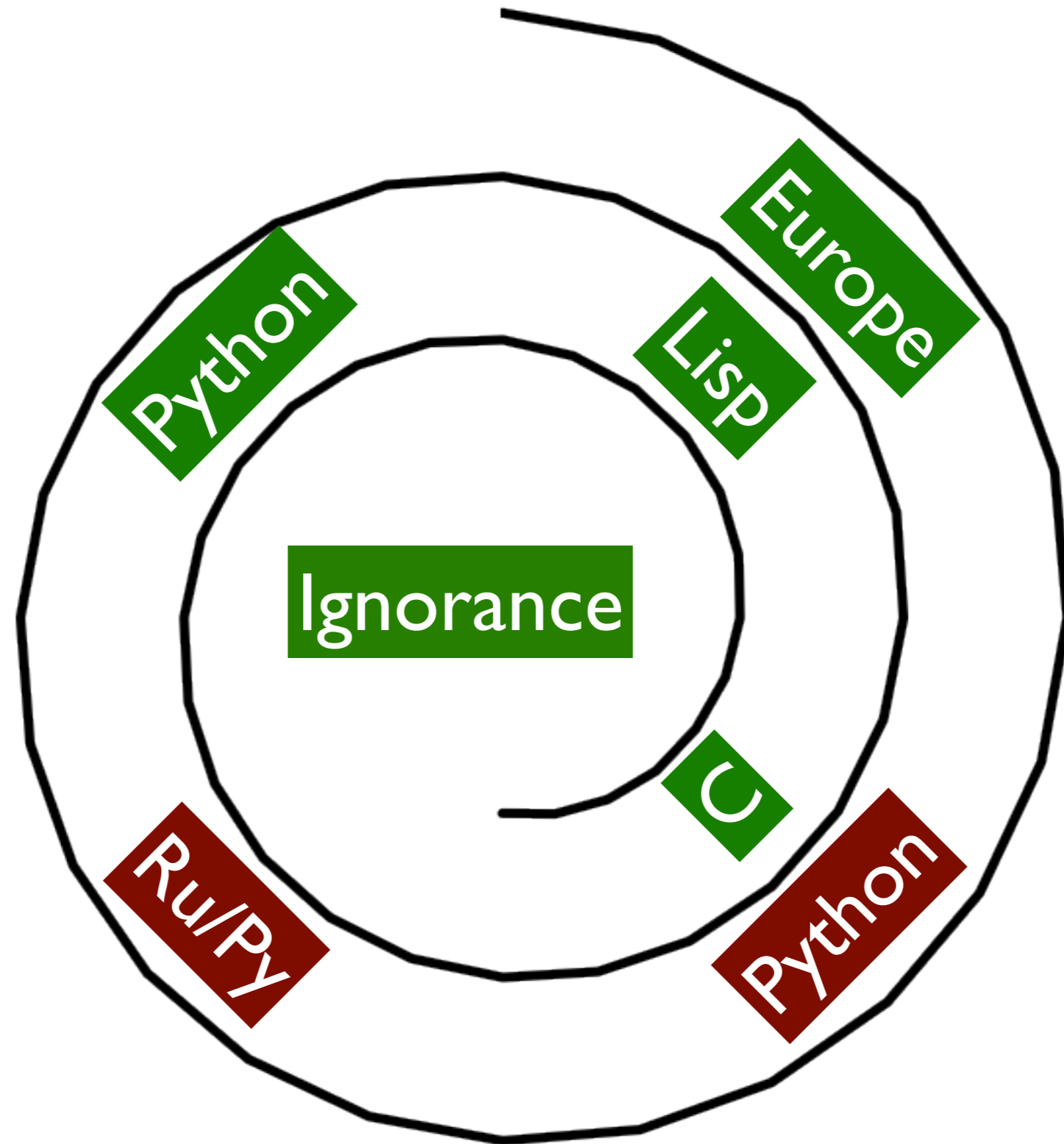

Ruby is the serious one?

"A Python programmer rejects a new idea without considering its value. A Ruby programmer accepts a new idea without considering its value."



Q3-Q4 2010





January 1, 2011

SCREENCASTS FOR SERIOUS DEVELOPERS

Destroy All Software screencasts are short: **10 to 15 minutes**, but dense with information. They're released each week, covering advanced topics like these:

Unix: combining the pieces; using the entire Unix operating system as your IDE

Fast, powerful test suites: achieving **1ms per test**

Fully embracing the power and danger of dynamic languages (focusing on **Ruby**)

Test-Driven Development (**TDD**): real-world implications and the relationship between TDD and **OO design**

Git & DVCSes — rebasing safely, mining statistics, and customizing the interface

Using **Vim** faster than the next guy

[Subscribe Now →](#)

Instance Variables

```
# Ugly 1
```

```
class Horse
```

```
  def what
```

```
    5
```

```
  end
```

```
end
```

```
puts Horse.new.what # => 5
```

```
# Ugly 2
```

```
class Horse
```

```
  def what
```

```
    @mustard
```

```
  end
```

```
end
```

```
puts Horse.new.what # => nil
```

```
# Ugly 3
```

```
class Horse
```

```
  def what
```

```
    @mustard = 5
```

```
  end
```

```
end
```

```
puts Horse.new.what # => 5
```

```
# Ugly 4
```

```
class Horse
```

```
  def what
```

```
    @mustard ||= 5
```

```
  end
```

```
end
```

```
puts Horse.new.what # => 5
```

```
# Ugly 4
```

```
class Horse
```

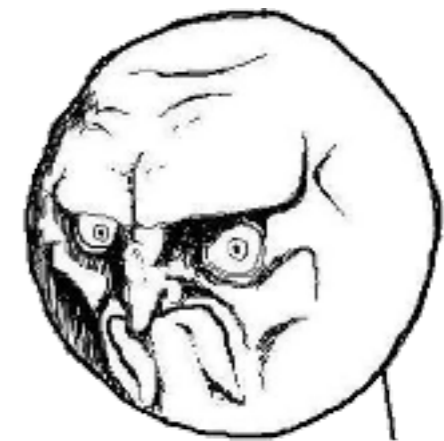
```
  def what
```

```
    @mustard ||= 5
```

```
  end
```

```
end
```

```
puts Horse.new.what # => 5
```



NO.

```
# Ugly 5
```

```
class Horse
```

```
  def expensive
```

```
    @expensive ||= compute_it
```

```
  end
```

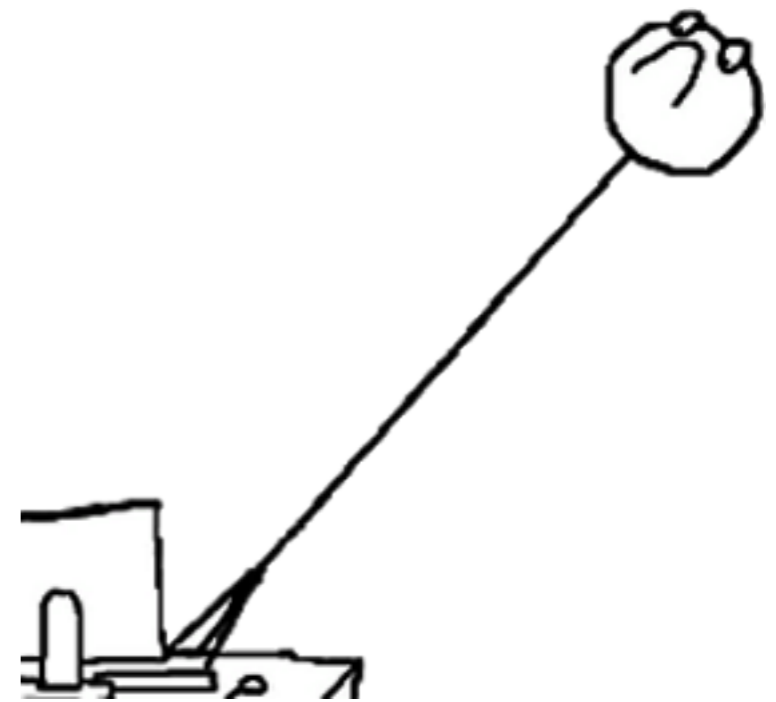
```
end
```

```
puts Horse.new.expensive
```

```
# Ugly 5
```

```
class Horse
  def expensive
    @expensive ||= compute_it
  end
end

puts Horse.new.expensive
```



"One and only one"

```
# generator
```

```
objects.each do |o|
```

```
  yield o
```

```
end
```

```
# list comprehension
```

```
map do |obj|
```

```
  obj.name
```

```
end
```

```
# decorator
```

```
my_decorator do
```

```
  do_decorated_stuff
```

```
end
```

```
# context manager
```

```
resource do |r|
```

```
  do_stuff_with(r)
```

```
end
```

```
# generator
```

```
x do |y|
```

```
  yield y
```

```
end
```

```
# list comprehension
```

```
x do |y|
```

```
  z
```

```
end
```

```
# decorator
```

```
x do
```

```
  z
```

```
end
```

```
# context manager
```

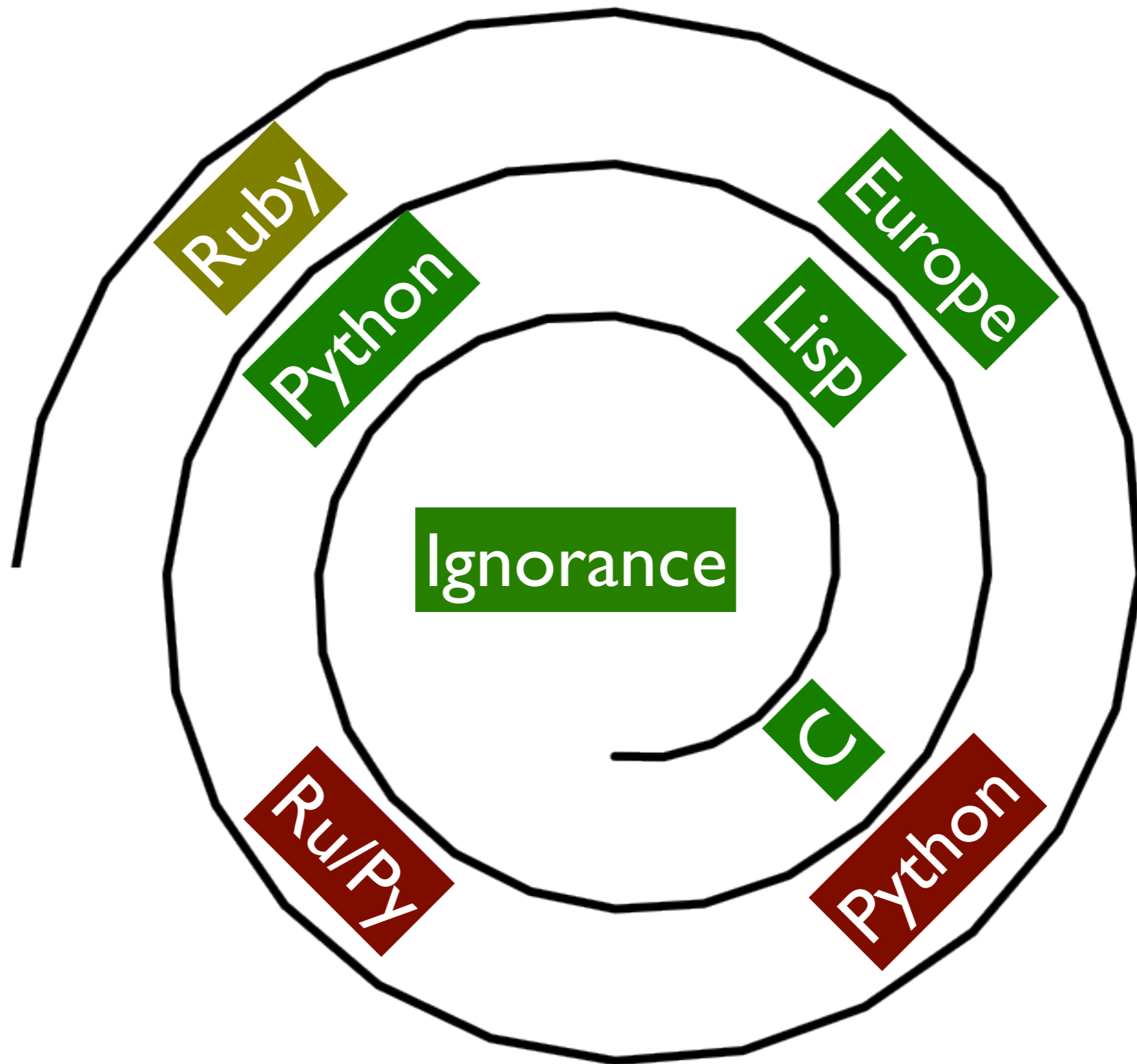
```
x do |y|
```

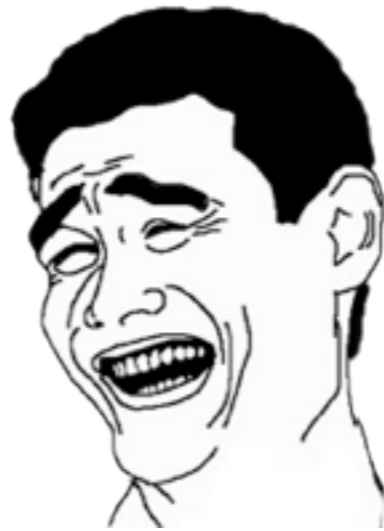
```
  z
```

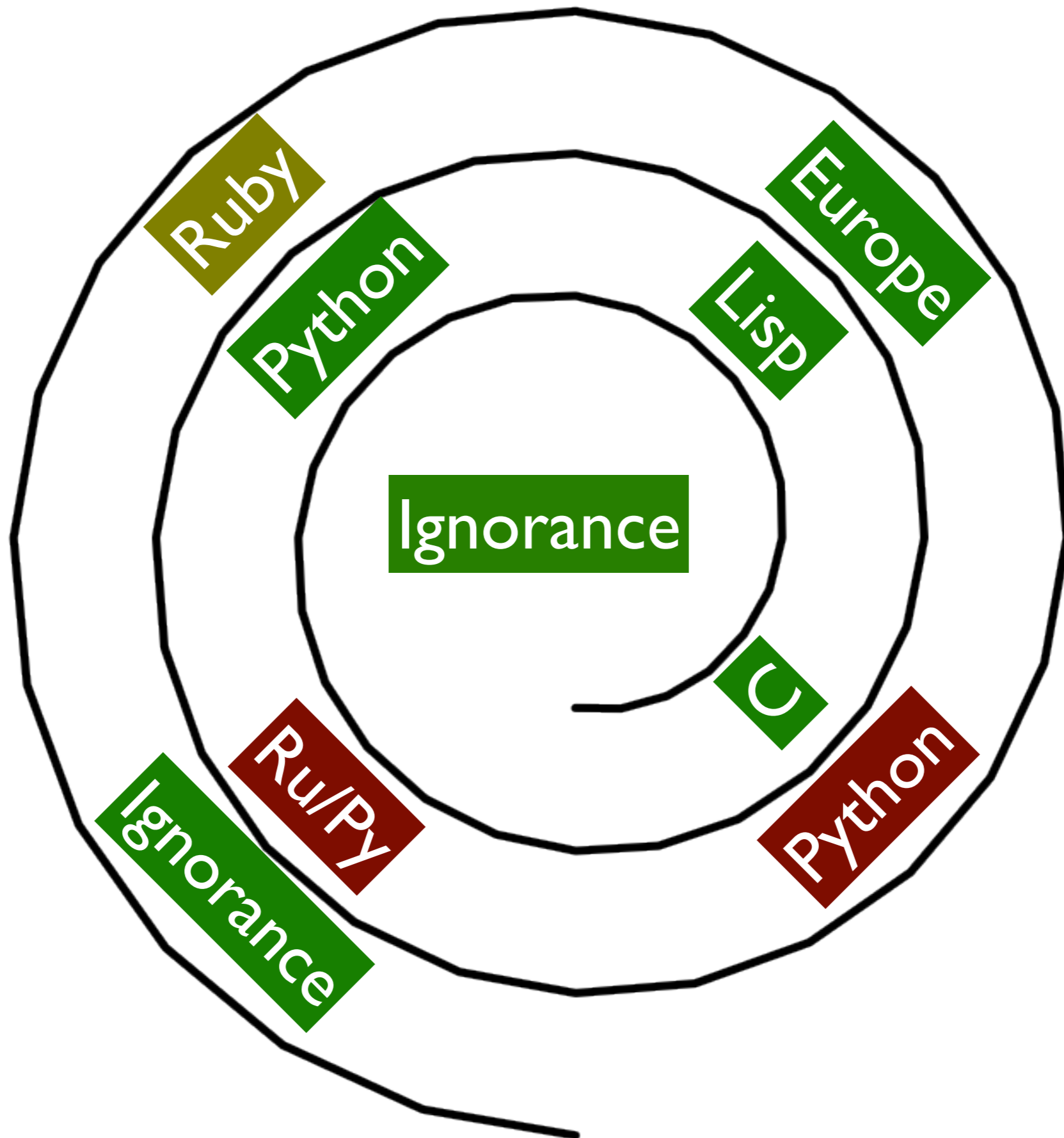
```
end
```

Generators
Comprehensions
Decorators
Context Managers

Which language is complicated?







Empirically:

Ruby's complexity isn't confusing

Ruby's community is frustrating

Ruby's tools are more modern

**Ruby's culture is more serious
about testing and design**

Why does it matter?

My opportunity was rare

Blocks are the key to happiness

@garybernhardt

destroyallsoftware.com