

The Higher Order Rubyist

Robert Pitts

@rbxbx

<http://regretful.ly>

Why Bother?

Solution?

The tools that shall aid our pursuit

batteries which have been included...

Enumerable

(um, obviously?)

All your favorites HOFs

(well, maybe just some)

`map | reduce | filter :: etc!`



Deconstructing


```
1 # somewhat contrived example code, imagine that
2 people = { "Robert" => ["Male", 25, "Stupid Software Person"],
3             "Lindsey" => ["Female", 25, "Illustrator"] }
4
5 names_ages_and_careers = people.map do |name, (_, age, career)|
6     [ name, age, career ]
7 end
8 names_ages_and_careers
9 => [["Robert", 25, "Stupid Software Person"], ["Lindsey", 25, "Illustrator"]]
```

First-ish-class Functions



blocks

procs

lambdas

methods (oh my)

Proc.curry

Currying & Partial Application

```
1 add = ->(x,y) { x + y }  
2 # currying  
3 make_adder = add.curry  
4 # partial application  
5 add_two = make_adder.call(2)  
6 add_two.call(40)  
7 => "Life, the Universe and Everything"
```

Continuations

via `callcc`
(hello scheme!)

We even fixed all (most) of the
horrible memory leaks!

**But maybe for not much
longer :(**

**RUBY CONTINUATION CONSIDERED
HARMFUL -- Koichi Sasada**

TCO

As of Ruby 1.9 we can recurse without fear.

TCO

```
1 RubyVM::InstructionSequence.compile_option = {  
2   :tailcall_optimization => true,  
3   :trace_instruction => false  
4 }
```


So that's what Ruby has to offer us natively, but surely there's more?

**You may be surprised to
learn this...**

**Any sufficiently
complicated program
contains an ad-hoc,
informally-specified, bug-
ridden, slow
implementation of half of a
Functional language.**

Generalized Greenspun Rule

Hamster



- persistent
 - efficient
- immutable
- thread-safe
 - lazy

list | stack | queue | hash | set | vector

```
1 require 'hamster/hash'
2 simon = Hamster.hash(first_name: "Simon", last_name: "Harris")
3 # non destructive, creates a copy
4 simons_brother = simon.put(first_name: "Rich")
5
6 simon
7 => { first_name: "Simon", last_name: "Harris" }
8 simons_brother
9 => { first_name: "Rich", last_name: "Harris" }
```

Stunted

defn

hooray, bags of functions instead of
bags of methods

Ruby#Facets

frequency | zip_map | mash

Celluloid

Not strictly Functional

Actors

People even use it in production!

M-m-m-m-m-onads!

I feel like if this weren't Strange Loop people would start giggling at the mere mention of the word.

Ick

```
1 # block-local variable binding
2 let(fetch_person_from_db) do |person|
3   person.do_something
4 end
5 # if you squint hard enough, the maybe monad
6 maybe(fetch_person_from_db) { |person| person.manager.authority_level.permissions }
```

do_notation

<http://skillsmatter.com/podcast/home/november-lrug>

```
1 value = Maybe.run do
2   x <- 1
3   y <- nil
4
5   x + y
6 end
7
8 value == nil
```

Why stay with Ruby?

Silly Haircuts

and a lack of scientific rigor

Thanks rails

Diversity!

(Sorry Haskell/Qi)

Thanks rails

Legacy

Ew, dirty word

Y Not?

h/t to Jim Weirich

Rubyist?

Non-Rubyist?

Function λ

