The Legacy of Jim Weirich

https://github.com/mdoel/legacy_of_jim

Mike Doel
VP Engineering, Neo
@mikedoel

Eulogy

~~Eulogy~~

1. Read the classics

Timeless?

Can a programming book stay relevant?

- Software Tools - 1976
- SICP - 1985
- Introduction to Algorithms - 1990
- Design Patterns - 1995
- Refactoring - 1999

10 Papers

(really fast)

Jim Weirich
Chief Scientist / EdgeCase
jim@edgecase.com
@jimweirich

EdgeCase
software artisans

https://github.com/jimweirich/presentation_10papers

10 Papers

- On the criteria to be used in decomposing systems into modules – Parnas
- A Note On Distributed Computing – Waldo, Wyant, Wollrath, Kendall
- Can Programming Be Liberated from the von Neumann Style? – Backus
- Reflections on Trusting Trust – Thompson
- A Laboratory For Teaching Object-Oriented Thinking - Beck, Cunningham
- 5 more…

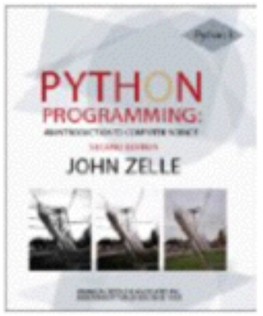
Best Sellers in **Computer Programming**

1. 

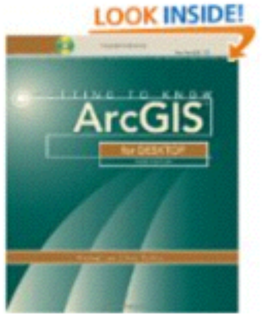
HTML and CSS: Design and Build Websites
by Jon Duckett
(582)
Paperback
$17.39
140 used & new from $13.74

2. 

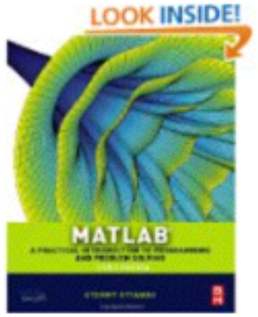
Cracking the Coding Interview: 150 Pr...
by Gayle Laakmann McDowell
(354)
Paperback
$26.93
72 used & new from $12.40

3. 

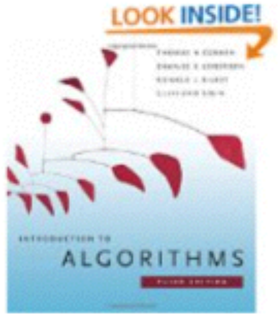
Python Programming: An Introduction t...
by John M. Zelle
(84)
Paperback
$19.99
95 used & new from $19.99

4. 

Getting to Know ArcGIS for Desktop
by Michael Law
(34)
Paperback
$48.88
141 used & new from $42.37

5. 

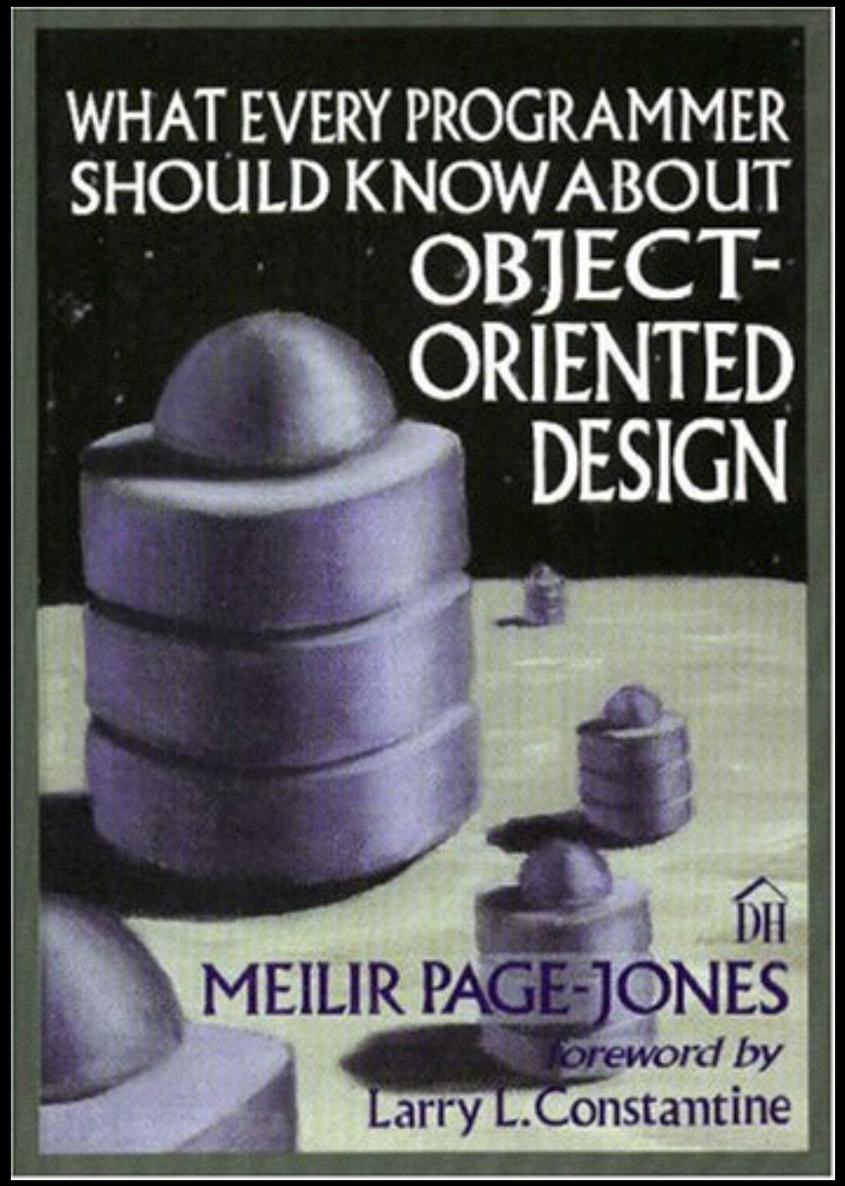
Matlab, Third Edition: A Practical In...
by Stormy Attaway
(23)
Paperback
$33.81
102 used & new from $29.38

6. 

Introduction to Algorithms, 3rd Edition
by Thomas H. Cormen
(181)
Hardcover
$75.87
136 used & new from $58.98

1. Read the classics

2. Apply the ideas


WHAT EVERY PROGRAMMER SHOULD KNOW ABOUT
OBJECT-ORIENTED DESIGN
DH
MEILIR PAGE-JONES
foreword by
Larry L. Constantine

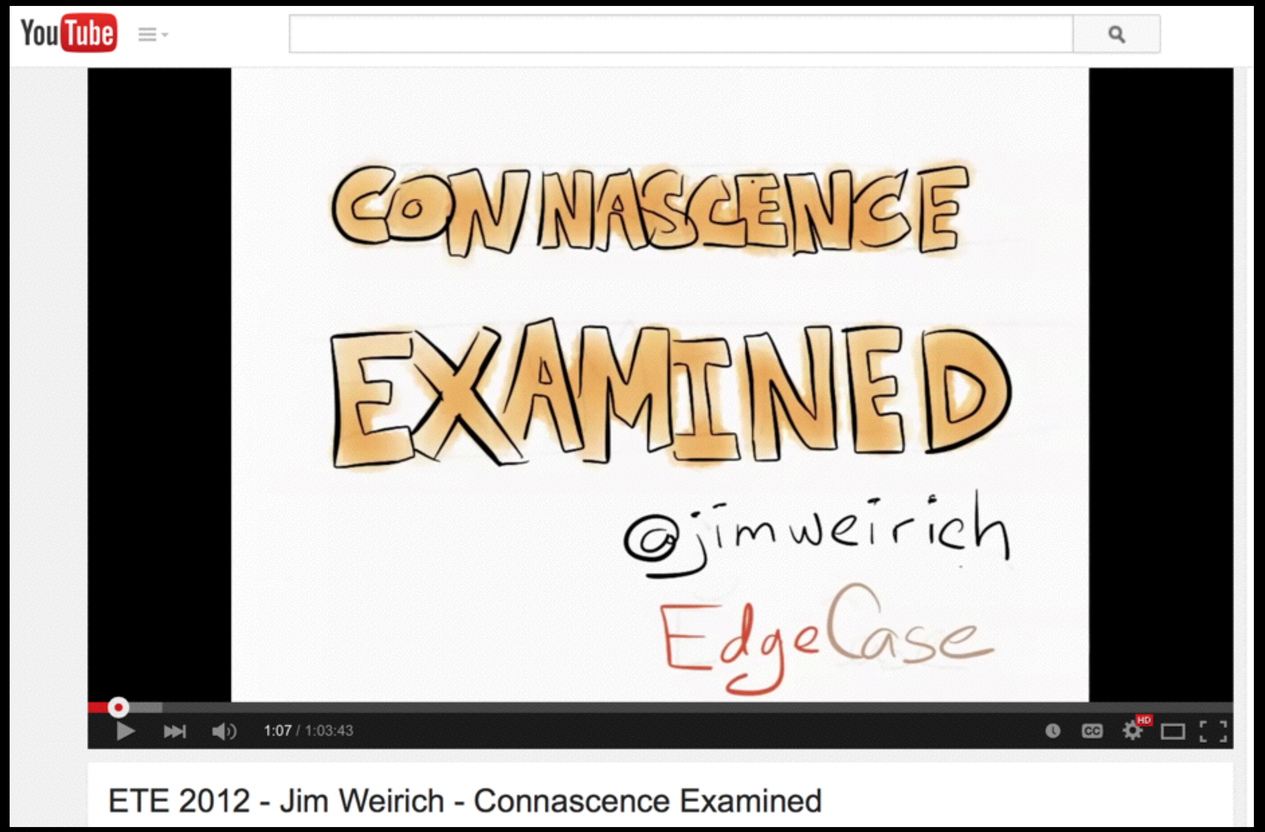
Connascence

The common birth of two or more at the same time; the production of two together.

That which is born or produced with another.

The act of growing together.

Emerging Technologies for the Enterprise 2012 Conference



https://www.youtube.com/watch?v=HQXVKHoUQxY

Connascence

Name

Position

Meaning

Algorithm

Type

Execution

Timing

Value

Identity

Connascence

Name

Position

Meaning

Algorithm

Type

Execution

Timing

Value

Identity

```
def foo
  …
end



def bar
  …
  foo
  …
end
```

Connascence

Name

Position

Meaning

Algorithm

Type

Execution

Timing

Value

Identity

```
def gsub(pattern,replacement)
  …
end



def bar
  …
  str.gsub(/\D/,'')
  …
end
```

Connascence

```
window.addNewControl("Title", 20, 50, 100, 50, true);
```

Connascence

```
window.addNewControl("Title", 20, 50, 100, 50, true);


[window addNewControlWithTitle:@"Title"
                       xPosition:20
                       yPosition:50
                           width:100
                          height:50
                      drawingNow:YES];
```

Connascence

Name

Position

Meaning

Algorithm

Type

Execution

Timing

Value

Identity

```
Fixnum.instance_methods true

Fixnum.instance_methods false
```

Solid Ruby - RubyConf 2009

http://www.confreaks.com/videos/185-rubyconf2009-solid-ruby

“As a presenter, Jim was a master of helpful metaphors. I often feared when he started a presentation that the metaphor would be too forced and come off hokey. It never did. He was able to take any complicated thing and explain it in a simple way by matching it to the right metaphor.”

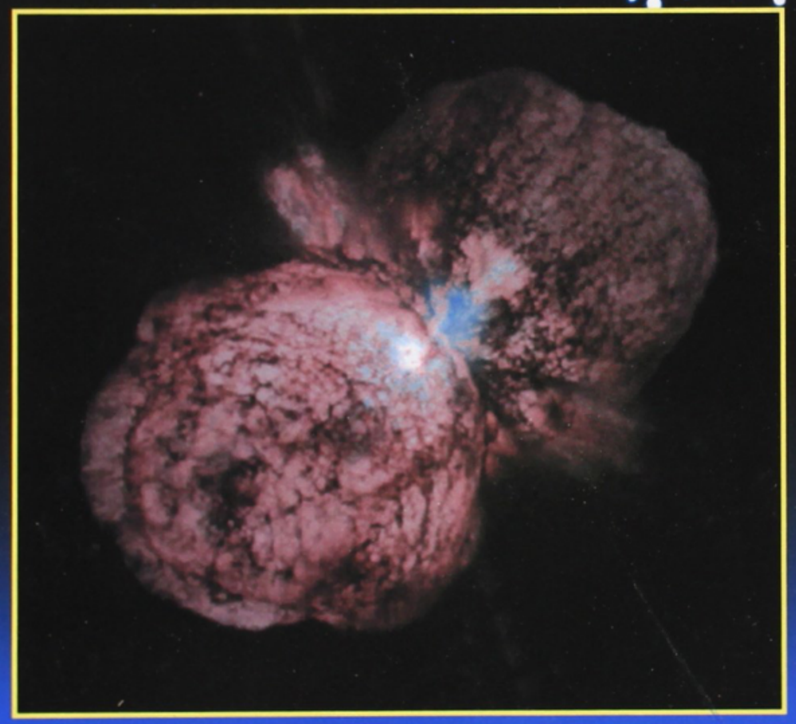
- Chad Fowler

“I always watched his public speaking because I wanted to learn from him. He knew how to turn complex topics in to something that was easy to understand for anyone. He knew how to tell stories, and I hope that someday I can do it as well as he did.”

- Aaron Patterson

AGILE SOFTWARE DEVELOPMENT

Principles, Patterns, and Practices



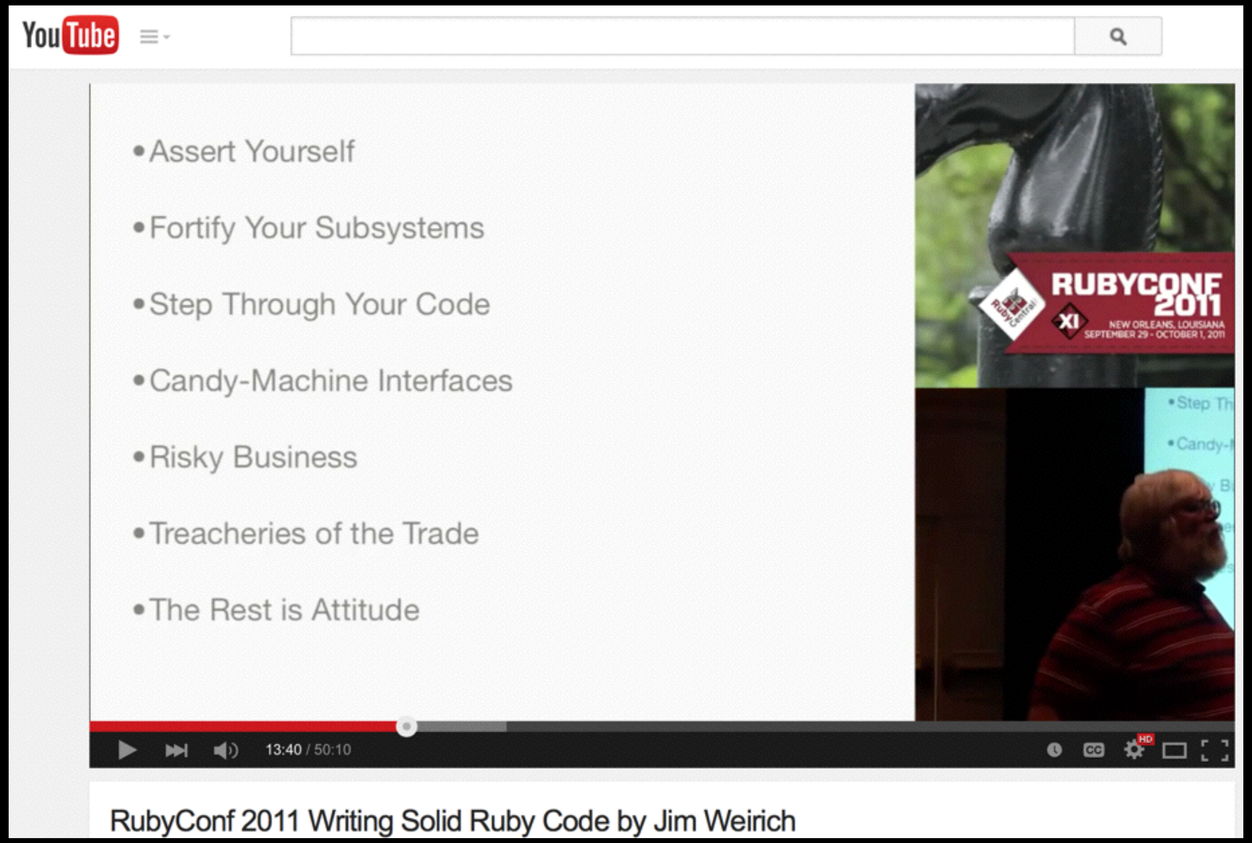
Robert C. Martin

with contributions by James W. Newkirk and Robert S. Koss

SOLID

- S - Single Responsibility Principle
- O - Open/Closed Principle
- L - Liskov Substitution Principle
- I - Interface Segregation Principle
- D - Dependency Inversion Principle

Writing Solid Ruby Code - RubyConf 2011



https://www.youtube.com/watch?v=FR95rp-9Oo4

Software Tools - Kernighan & Plauger

Object Oriented Software Construction - Bertrand Meyer

Refactoring - Martin Fowler

Extreme Programming Explained - Kent Beck

Writing Solid Code - Steve Maguire

2. Apply the ideas

TESTS

MAKE SURE THEY'RE ALL GREEN OR ELSE

3. Understand and be respectful of history

“C++ was designed to provide Simula’s facilities for program organization together with C’s efficiency and flexibility for systems programming. It was intended to deliver that to real projects within half a year of the idea. It succeeded.”

- The Design and Evolution of C++

“Even when we were neck deep in some mundane project that nobody was enjoying, while others (probably mostly me) were complaining, Jim would not only cheerfully do that work but he would try make the people around him happier as well. In the six years I worked beside him, and the couple years I knew him before that, I honestly can't remember a time where Jim was angry, or upset, or even being negative.”

- Scott Barron

3. Understand and be respectful of history

4. Be humble

Why rake?

Ok, let me state from the beginning that I never intended to write this code. I'm not convinced it is useful, and I'm not convinced anyone would even be interested in it. All I can say is that Why's onion truck must by been passing through the Ohio valley.

What am I talking about? … A Ruby version of Make.

See, I can sense you cringing already, and I agree. The world certainly doesn't need yet another reworking of the “make” program. I mean, we already have “ant”. Isn't that enough?

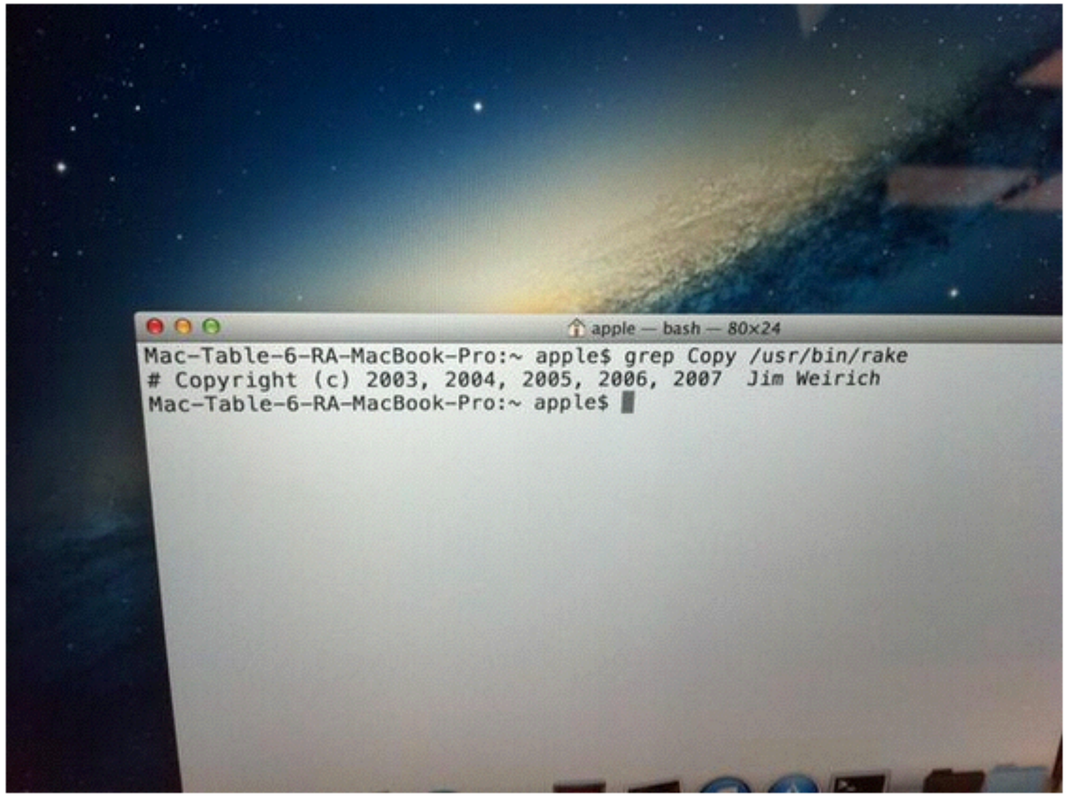
https://github.com/ruby/rake/blob/master/doc/rational.rdoc

Jim Weirich
@jimweirich

Following

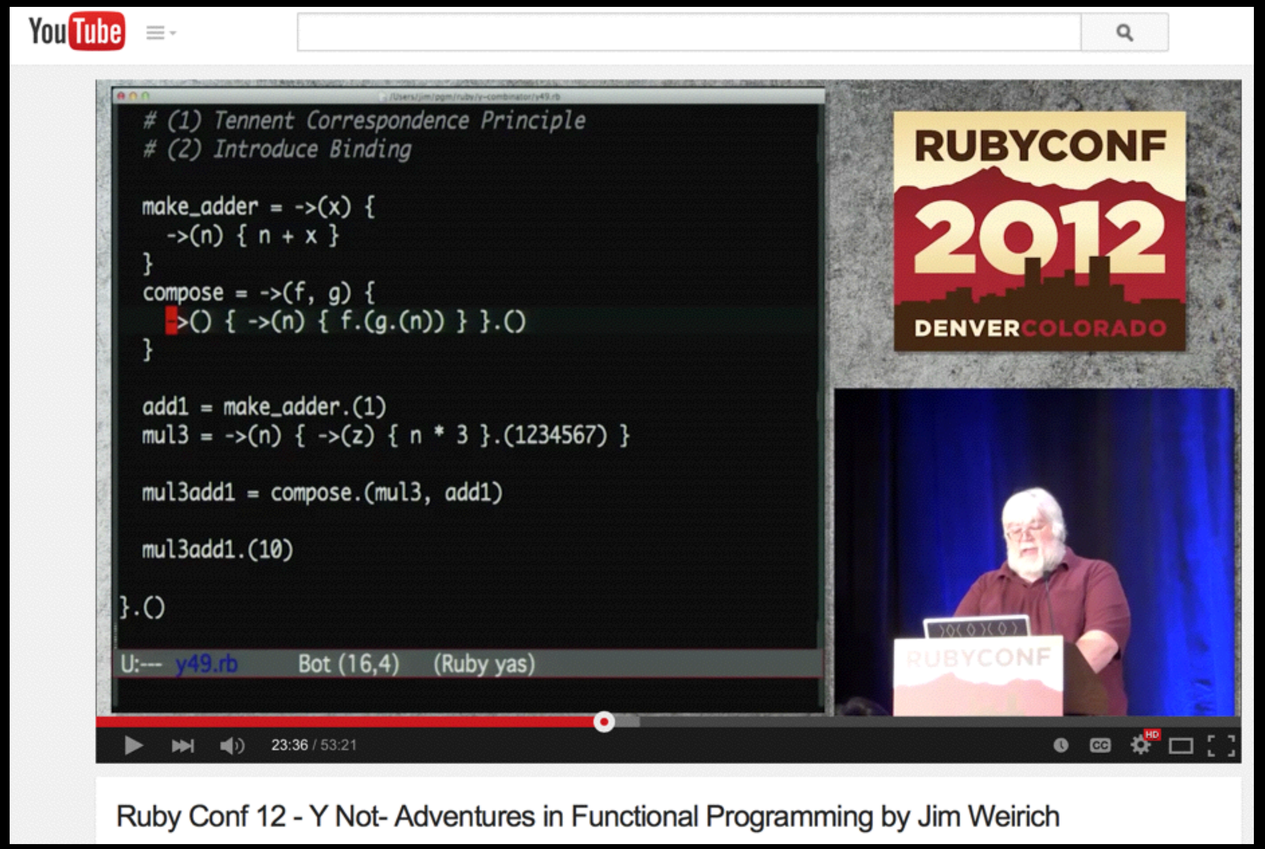
My favorite thing to do in the apple store:
pic.twitter.com/ZeBHgKxmNt

Reply Retweet Favorite More



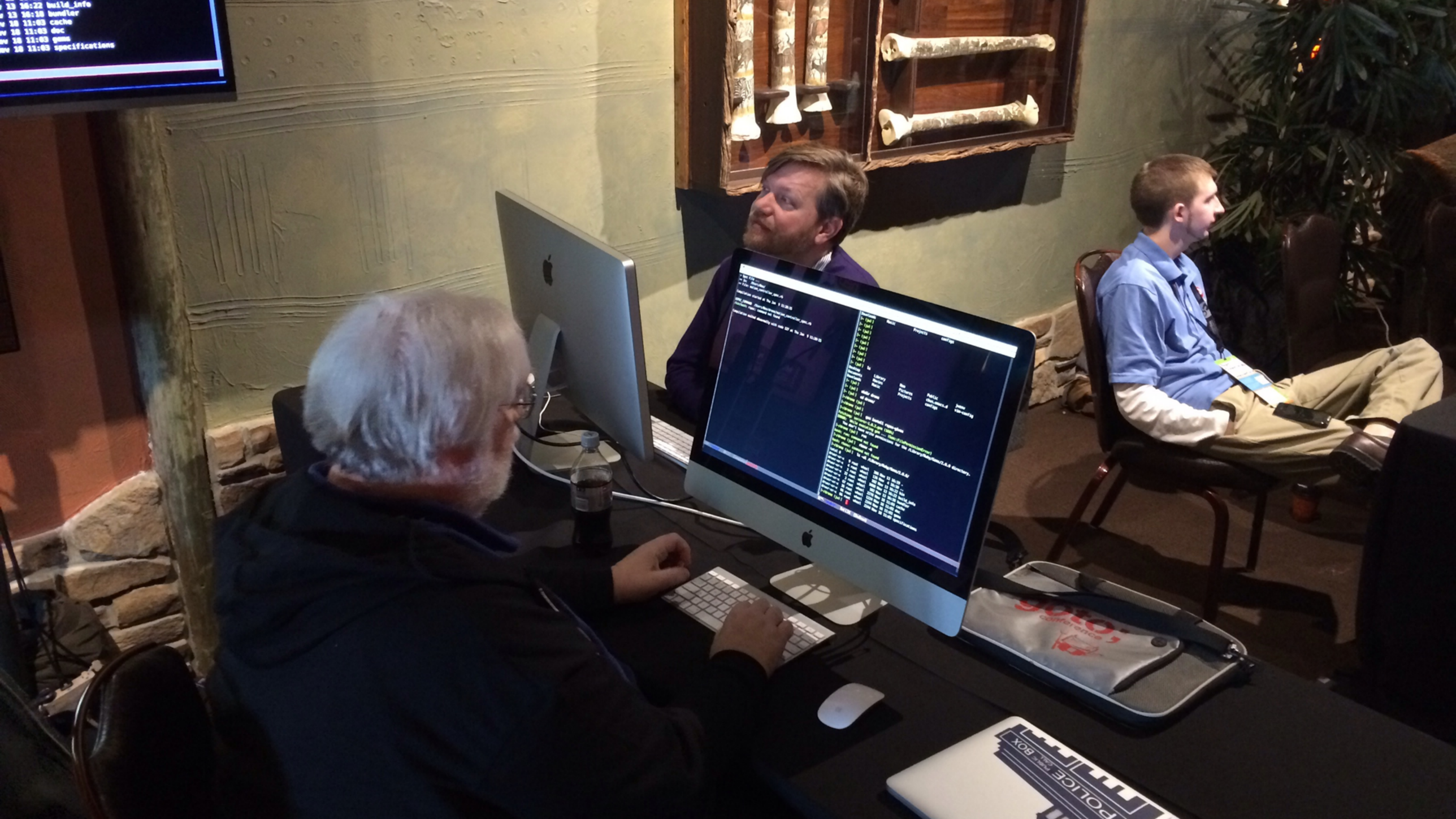
5. Practice because you love it

Y Not - Adventures in Functional Programming - RubyConf 2012



https://www.youtube.com/watch?v=FITJMJjASUs

- Released more than 50 albums
- Seven time recipient of GMA Dove Award
- Frequently listed as one of the top three “finger style” guitarists by Guitar Magazine polls




Welcome to
Kaleban
REGISTRATION


We Make.
We Validate.
JVC

“Jim taught me that magic isn’t a dirty word in programming. It’s a sense of wonder and an expansion of what you thought was possible.”

- David Heinemeier Hansson

5. Practice because you love it

Jim W. Ok, I get these emails occasionally. Thought you might enjoy it.

View paste

My name is Terry Thomas. I would like to know if you have Power Rake Available and i'm looking for less expensive ones you have. Could you please let me know the prices that are
Power Rake
I would be happy to make my payment by credit card if you accept them.

I await your reply,
Sincerely,
Terry Thomas.

6. Live your faith


NEW BURLINGTON
CHURCH OF CHRIST

Jim Weirich
@jimweirich

Following

Got to play silent night on guitar at church this morning (Tweeted for the benefit of @Jamis :)

12:38 PM - 21 Dec 2008

“He never had to be right.”

- Ken Barker

6. Live your faith


MEN
CodeMash
v2.0.1.4
BUILD THE RIGHT THING.

7. Teach

The Weirich Fund

Sharing Jim’s passion through education

http://jim.neo.com

Gracie Carver-Dews
Xavier University

Tyler Parcell
University of Cincinnati

The Legacy of Jim Weirich

1. Read the classics
2. Apply the ideas
3. Understand and be respectful of history
4. Be humble
5. Practice because you love it
6. Live your faith
7. Teach

The Weirich Fund

Sharing Jim’s passion through education

http://jim.neo.com