Ruby
Past, Present and Future

Yukihiro "Matz" Matsumoto
まつもと ゆきひろ
matz@ruby-lang.org

thou

The News in 2004-2006

Ruby on Rails

Ruby on Rails

- Changed Our Lives a Lot
- Changed Ruby

The News in 2007-2008

Alternative Implementations

- YARV
- JRuby
- Rubinius
- Ruby.NET
- IronRuby

Alternative Implementations

- Complete
  - Run Rails
- Faster
  - Than Plain 1.8 (MRI)

But Look at Bright Side

I consider myself as

- No Great Programmer
- But Language Designer

So..

I can Rather Focus on Language Issues

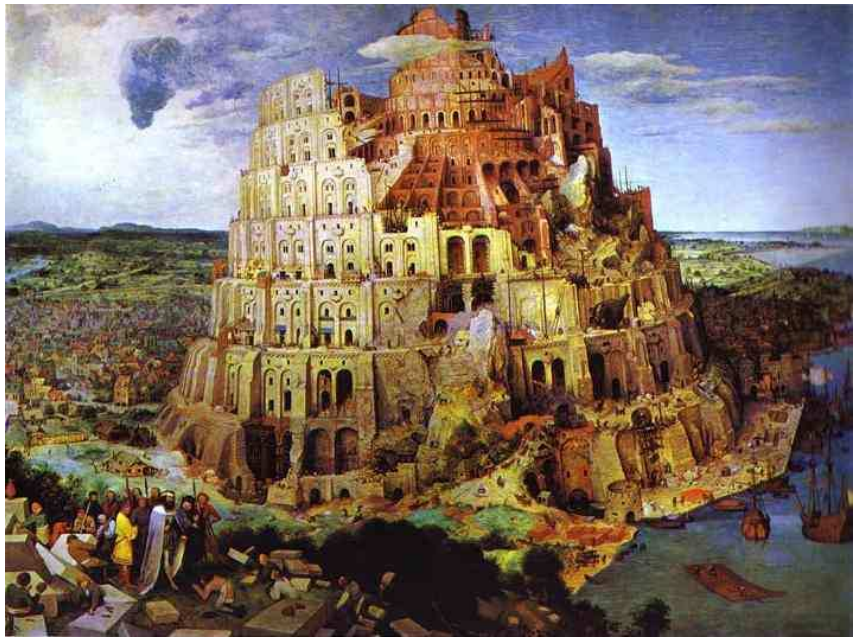
- Not Implementation
- But I Have Mixed Feeling

Past, Present and Future

- History
- Philosophy
- Forecast

The History

More than 4,000 years ago



We had One True Language.

Seek for True Language

Fortran

Seek for True Language

Cobol

Seek for True Language

Lisp

Seek for True Language

Algol

Seek for True Language

C / C++

Seek for True Language

Java / C#

Seek for True Language

Scripting

Seek for True Language

Perl

Seek for True Language

Python

Seek for True Language

Ruby

Pre-History

- OO Fanboy
- Language Geek

In 1993

- Project Started
  - 1993-02-24
- Mere Hobby

Goals

- Scripting
  - a la Perl
- Nice Clean Syntax
- With OO

Real Goal

To Enjoy

- Making Language
- Implementation
- Programming

Design Process

- Lisp Sentatics
- Smalltalk OO
- Conservative Syntax

Design Process

- Deconstruct Perl
- Reorganize into Class Library

Design Process

- Iterators from CLU
- Higher-order Functions

Design Process

- Some Spice from Python
- ..and Others

Released

1995-12-21

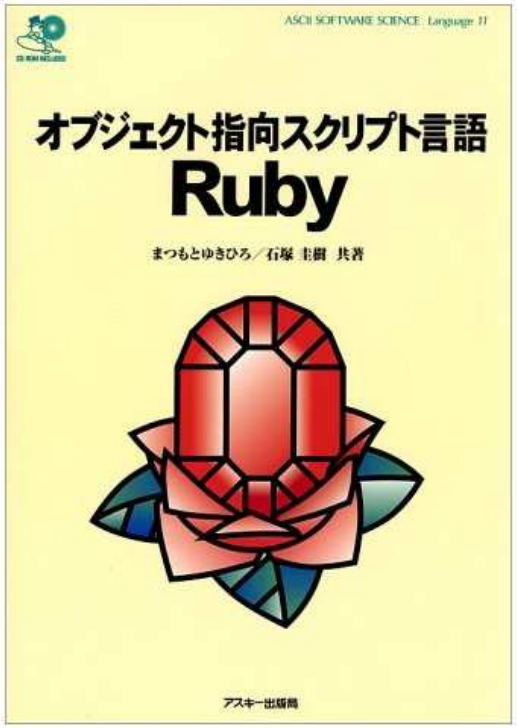
fj.sources

In 1997

- Hired by NaCl
- Full-time OSS Developer

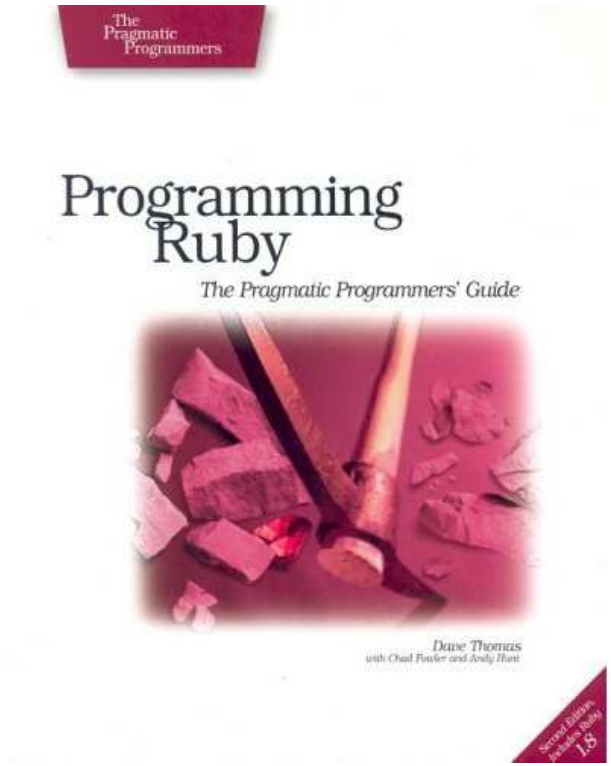
In 1999

First Book



In 2000

First English Book



In 2004

Ruby on Rails

10 Years Ago

- Ruby?
- What's That?
- Language?
- See This Cool Java!

5 Years Ago

- Ruby?
- I’ve Heard of It.
- But I haven’t Used It YET.

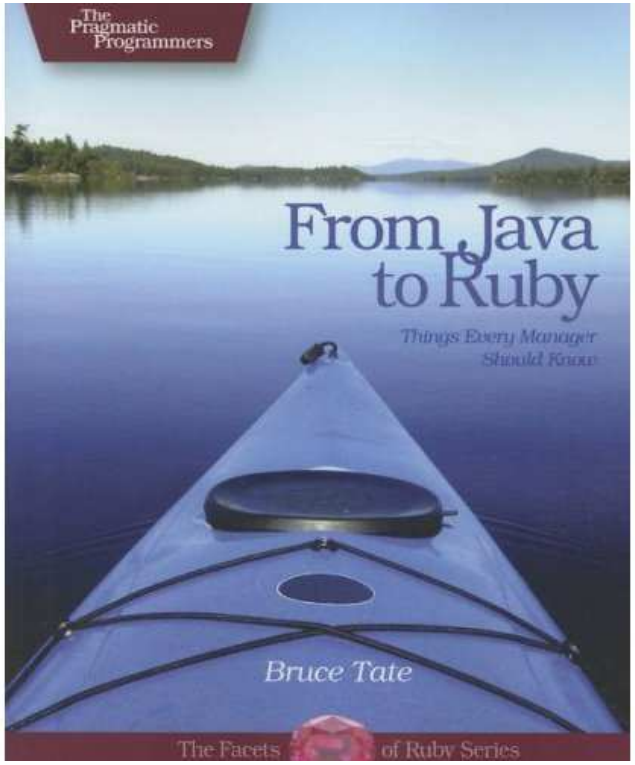
2 Years Ago

- Ruby!
- I Know!
- It’s for Rails, Right?

Status Quo

Ruby's Mindshare

From Java To Ruby



Why Ruby?

- Rails!
- Productivity
- Joy

No Need to Explain

Productivity

More and More Emphasized Recently:

- IT Market Grows
- Budget Don’t
- Tigher Schedule

How to Be Productivity

- Tool Support
  - IDE, etc.
- Methodology
  - XP, Scrum, etc.
- Language
  - Ruby!

Language for Productivity

- A Language is one of many tools
- But it changes project's characteristic
- Sapir-Whorf hypothesis

Sapir-Whorf hypothesis

Language determines the way we think.

Sapir-Whorf hypothesis

A Language Can Make Us Feel

- More Powerful
- More Effective
- More Freedom

Joy

We Can Feel Joy Thrugh Programming

- Ruby’s Primary Goal
- Too Underestimated in the Past

Ruby Became "Enterprisey"

a lot of big names using Ruby (and Rails)

- Sun
- Microsoft
- Oracle
- IBM
- etc.

Is It GOOD?

Probably

- We can Earn Bread
- We can Meet Here
- But not the First Priority
  - Programmer First

Is It DANGEROUS?

Maybe

- Ruby is not Rails
- Rails Gathered Lot of People Who Don’t Know About Ruby

Ruby has Longer Lifecycle

- 15 Year-old Program
- But Some Languages are more than 50!
- We Must Survive

Status Quo

- Implementation
- Language
- Unicode

Implementation

- MRI (1.8)
- JRuby (1.8)
- Rubinius (1.8)
- YARV (1.9)

MRI

- Matz’s
- Ruby
- InterPreter

JRuby

Ruby on JVM

- 1.8 Compatible
- Faster Than MRI
- Sometimes Beats 1.9

Rubinius

Ruby in Ruby

- Core in C
- Library in (subset of) Ruby
- Progressing Fast

YARV

Yet Another Ruby VM

- Bytecoded Stack VM
- Fastest Ruby on Earth
- 50 Times Faster on Some Benchmarks
- Can be Even Faster

Language

1.9 is Bleeding Edge

- Clarifying Edges and Corners of the Language
- Adding New Features

See Google TechTalk Video on YouTube

1.9 Significants

- Enumerator
- Fiber
- Block Scope
- M17N

Enumerator

- Built-in
- Enumerator Chain
- External Iterator

Enumerator Chain

```
ary.map.with_index{|x,i|
  ...
}
ary.find.with_index{|x,i|
  ...
}
```

External Iterator

```
e1 = [1,2,3,4].each
e2 = [10,11,4].each
loop {
  p e1.next+e2.next
}
# prints 11, 13 and 7
```

Fiber

Cooperative Thread

- Switch Context Explicitly
- Used to Implement External Iterator
- More Lightweight Than Threads

Multilingualization

M17N means a lot

- Handle Locales
- Handle Characters
- Handle Glyphs
- Handle Cultures

M17N in Ruby

- Handle Locales
- Handle Characters
- Handle Glyphs
- Handle Cultures

How to Handle Characters

- Ignorance
- Unicode
- Multi-Encoding

How to Handle Characters (1)

By Ignoring Tough Cases

- ASCII
- ISO-8859-1 (Latin-1)
- Not Good for Japanese

How to Handle Characters (2)

By Using Unicode

- UTF-8
- UTF-16
- UTF-32
- Most Languages Do This
  - Java, Perl, Python...

UTF-8

- ASCII compatible
- Variable Char Length
- O(n) for Random Access
- Perl uses UTF-8

UTF-16

- 16bits Per Character
- Variable Char Length
- O(n) for Random Access
- Requires More Space
- Endian Aware
- Java, Python use UTF-16

UTF-32

- 32bits Per Character
- Fixed Char Length
- O(1) for Random Access
- Much More Space
- Endian Aware
- A Few Languages use UTF-32

Unicode Problems

- No Perfect Encoding
- No Perfect World
- Conversion Issues
  - Performance Cost
  - Broken Characters
- Extra-Unicode Chars

If There's No Perfect Encoding

How about Making it Extensible?

- No Need for Conversion
  - for Most of the Cases
- No Encoding Breakage
- Extra-Unicode Chars Can be Handled

Issues

- Possible?
- Performance?

Possible?

Yes,

- We did for long time for Regexp
- We Made it in 1.9

Performance?

Yes,

- Except for Random Access (O(n))
- But Other Encoding Too
- O(1) if Encoding is Fixed Width

Encoding-aware

```
# -*- coding: UTF-8 -*-
print "Hello まつもと\n"
```

Encoding-aware

```
open(path, "r:euc-jp") {|f|
    ...
}
```

The Future

Ruby 2.0

Innovation Bait

- Motivation Matters
- OSS Should Move Forward or Die
- We need Baits to go on
- Scalability is the Keyword

Scalability

- Data Size
- Number of CPU
- Program Size
- Team Size

Data Size/CPU Number

- Parallell
- Distributed
- Actor

Team/Program Size

- Keyword Arguments
- Selector Namespace
- Method Combination
  - Aspect Oriented
- More Functional Programming

Keyword Arguments

- Order Free
- More Reabable
- Objective-C

Selector Namespace

- Open Class
  - jcode.rb
- But Global Side-effect is Basically Bad
- So Statically Scoped Open Class

Method Combination

- Method Replacement Using alias May Conflict
- Allow pre and postHooks around Methods
- Can Be Stackable

More Functional Programming

More Abstraction

- LazyArray
- Delay/Force

Summary

- Ruby Celebrated Its 15th Birthday on 2008-02-24
- Ruby Focuses on Programmers, not Computers

Summary

- Ruby Has Established in IT Market
- That’s Fine
- But Don’t Forget the Primary Goal

Summary

- Multiple Interpreters
- They are All Great
- YARV is Improving
- So are Others
- That’s Fine

Summary

- The Future is in Scalability
- We Keep The Language Evolving
- Since Motivation Matters
- We Need to Move Forward or Die

One More Thing...

Do You Play Golf?

- Not a Game of Balls and Clubs
- a Game to Make Programs Shorter

Goruby

A Language Specialized for Golf

```
## specialized method
h          # => Hello World
## abbribiated access
p "1".t_i  # 1
```

Goruby

You can Play with type:

make golf

Have Fun!

Thank You