The Power of Ruby

Yukihiro "Matz" Matsumoto

matz@ruby-lang.org

thou

Moore's Law

The number of Transistors in LSI Doubles Every

18 Months

Moore’s Law Means:

Computer Grows Exponentially

- Faster
- Cheaper
- More Common

Faster Computer

PCs are Faster than Super Computers of 20 Years Ago

Cheaper Computers

We can Buy a PC for $400 Now

Common Computers

- Now Everyone Owns Computers
  - Personal Computers
  - Cell Phones

Cell Phone as a Computer

Cell Phone as a Computer

Everyone is Connected

- Broadband
- WiFi
- Mobile Networks

Influence in Programming

Moore’s Law Changes

- Software Complexity
- Programming Languages

Software Complexity

- No Business Can Be Run without Software
- We Need More Software
  - Quicker
  - Cheaper

Humans Don’t Improve

- Moore’s Law Does Not Apply to Humans

Productivity

- We Need More Software with Limited Resources

Productivity

- We Have Faster Computers
- Development Efficiency At the Cost of Runtime Efficiency

Productivity

- The Most Important Factor of Language Evolution
- Languages are One of the Tools for Productivity

How Languages Help Productivity

Sapir-Whorf hypothesis

Language determines the way we think.

Theorem #1

Languages influence human thought, more than you think

Programming Languages

Do programming languages influence human thoughts?

Thinking in Programming Language

- Natural languages are too ambiguous.
- Or, too verbose.
- Or, too indirect.

Thinking in Programming Language.

If programmers think in programming languages,
They must influence thoughts as much as
natural languages do.

Theorem #2

"languages" in Theorem #1 includes programming languages.

Sapir-Whorf in Languages

- BASIC programmers never use recursion.
- Lisp programmers use macros for everything.
- FORTRAN programmers can write FORTRAN program in any language.

Why don’t you choose a good language?

Programming langugages are so easy to learn.

What is a good language?

A language that

- helps human thought
- makes better programming experience

How to help thoughts

by providing computational models

- Procedural
- Object-Oriented
- Functional
- Etc.

How to help thoughts

by providing an interface to machines

- HI principle applies
- Usability matters

For Better Programming Experience

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

According to Dr. Jacob Nielsen

Learnability

How easy is it for users to accomplish basic tasks the first time they encounter the design?

Learnability

- Usability for Beginners
- Important to Acquire New Users
- "Common Sense" is the Key

Efficiency

Once users have learned the design, how quickly can they perform tasks?

Efficiency

- More important than learnability
- Efficiency is the top purpose of languages

Memorability

When users return to the design after a period of not using it, how easily can they reestablish proficiency?

Memorability

- Association
- Consistency
- Orthogonality
- Common Sense
- No Radical

Errors

How many errors do users make, how severe are these errors, and how easily can they recover from these errors?

Errors

- When you see repeated errors, you have to do something.
- Errors are the source of design inspiration.

Satisfaction

How pleasant is it to use the design?

Satisfaction

- We program to have fun.
- Even when we program for money, we want to have fun as well.

How Ruby Serves

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

How Ruby Serves

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Learnability

Ruby is very conservative except for a few places

- Quick to learn
- Quick to try

Learnability Example

Hello World!

```
print "Hello World\n"
```

Hello World

How Ruby Serves

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Efficiency

No Run-Time Efficiency

Ruby focuses on the cost of programming.

Devlopment Efficiency

Ruby focuses on the cost of programming by

- Simplicity
- Consistency
- Smartness

Simplicity

Do you like programming language to be simple?

- Probably you do.

Simplicity

Does language simplicity really help you?

- not always.
- need more complex tool sometimes

Need More Complex Tool

- Knife vs Chain Saw
- Bicycle vs Airplane

Human Heart: No Simple

- We love simplicity
- We love complexity
- We love easy problems
- We hate easy problems

Pseudo-Simplicity

Ruby is NOT a simple language.

Simplicity Example

Rakefile

- Rake = Ruby Make

```
task :default => [:test]

task :test do
  ruby "test/unittest.rb"
end
```

Simplicity Example

In Simpler Syntax

```
task({:default => [:test]})
task(:test, lambda(){
  ruby "test/unittest.rb"
})
```

Solution-Simplicity

Tool Complexity is OK
if it makes the Solution Simple

Efficiency Example

/bin/cat in Ruby

```
puts ARGF
```

It would be more than 50 lines of code in C

How Ruby Serves

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Memorability

- Conservativeness helps here too
- Easy-to-remember syntax
  - Ruby looks like other languages

Memorability Example

Can you write /bin/cat -n without looking anything?

I can, if I use Ruby.

```
ARGF.each_with_index{|line,i|
  printf "%4d %s",i,line
}
```

How Ruby Serves

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Errors

You will see less errors due to

- Consistent Syntax Rules
- Succinct Code
  - Less code, Less bug.

How Ruby Serves

- Learnability
- Efficiency
- Memorability
- Errors
- Satisfaction

Satisfaction

As a result, Ruby is fun to use.

It makes you feel smarter.

Few more tips in Design

- Succinctness
- Choosing Names

Succinctness is Power

- Less Code, Less Bugs
- Less Bugs, You Feel Yourself Smarter.
- You can be 10 times (or even 1000 times) more productive

Succinctness Example

```
class Sample {
  private static int fact(int n) {
    if (n == 1) return 1;
    return n * fact(n - 1);
  }
  public static void main(String[] argv) {
    System.out.println("6!="+fact(6));
  }
}
```

```
6!=720
```

Java

Succinctness Example

```
def fact(n)
  if n == 1
    1
  else
    n * fact(n - 1)
  end
end
print "6!=", fact(6), "\n"
```

```
6!=720
```

Ruby

Succinctness Example

```
def fact(n)
  (1..n).inject(:*)
end
print "6!=", fact(6), "\n"
```

```
6!=720
```

Ruby1.9

Less Restriction

```
print "200!=", fact(200), "\n"
```

```
200!=788657867364790503552363213932185062295
135977687173263294742533244359449963403342920
304284011984623904177212138919638830257642790
242637105061926624952829931113462857270763317
237396988943922445621451664240254033291864131
227428294853277524242407573903240321257405579
566022603190417032406235170085879617892222789
623703897374720000000000000000000000000000000
```

Ruby

Choosing Good Names

"Name" is the Power

- If you give a good name for a concept, 80% of the design is done already.

Ruby the Language

What’s Ruby?

- Open-source Language
- Object-oriented Language
- Scripting Language
- Glue Language

Open-Source Language

- Dual Licensed
  - GPL
  - Artistic-like

Object-Oriented Language

Ruby is object-oriented from the beginning.

Object-Oriented Language

Ruby = Smalltalk - Unfamiliar syntax

+ Perl’s scripting power
+ Python’s exception etc.
+ CLU’s iterator
+ a lot more good things

Ruby’s OOP Features (1)

Ruby is a Pure OOPL

- Everything = Object

Ruby's OOP Features (2)

Ruby is Even Purer than Java

- Every Class is an Object
- Every Procedure is a Method

Advanced OOP Features

- No Multi-Inheritance
- Mix-in
- Singleton
- Reflection

No Multi-Inheritance, but Mix-in

- Mix-in is as strong as multiple inheritance.
- But simpler.

Singleton

- Properties, especially methods belong to specific object.
- Prototype-based

Reflection

- Operation over meta information.
- Highly dynamic.

Scripting Language

Ruby is as Strong in Scripting as Perl.

Useful for scripting, but not limited to it.

Scripting Language

- built-in regular expression
- almost all equivalent functionality of core Perl

Scripting Language

- Access to all system calls on UNIX
- Access to all system calls on Win32

Glue Language

It’s pretty easy to make extension for Ruby.

Glue Language

- Raw API is Comprehensive.
- You can do almost everything in C too.
- SWIG is available.

Ruby is Good for

- Text Processing
- Web Programming
- XML Programming
- GUI Applications

Ruby is Good for

- Bioinformatics
- eXtreme Programming

"I love it. Conceptually it is really clean and sweet."

-- Kent Beck

Why I created Ruby

- Just for Fun
- Tool for me myself
- Ideal tool for Everyday Task

How I created Ruby

- Combine Good Things from the Past
- Design Conservative
- Design a Tool I Want to Use
- To Have Fun

The History of the Ruby Language

Pre-History

- OO Fanboy
- Language Geek

In 1993

- Project Started
- Mere Hobby

Goals

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

Real Goal

To Enjoy

- Making Language
- Implementation
- Programming

Process

- Lisp Semantics
- Smalltalk OO
- Conservative Syntax

Process

- Deconstruct Perl
- Reorganize into Class Library

Process

- Iterators from CLU
- Higher-order Functions using Blocks

Process

- Some Spice from Python
- ..and other languages

Released

1995-12-21

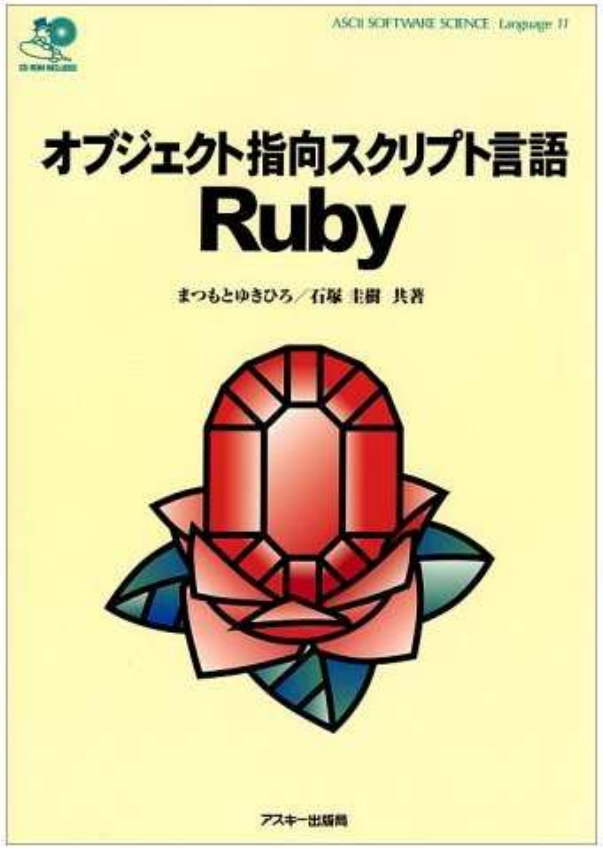
fj.sources

In 1997

- Hired by NaCl
- Became Full-time OSS Developer

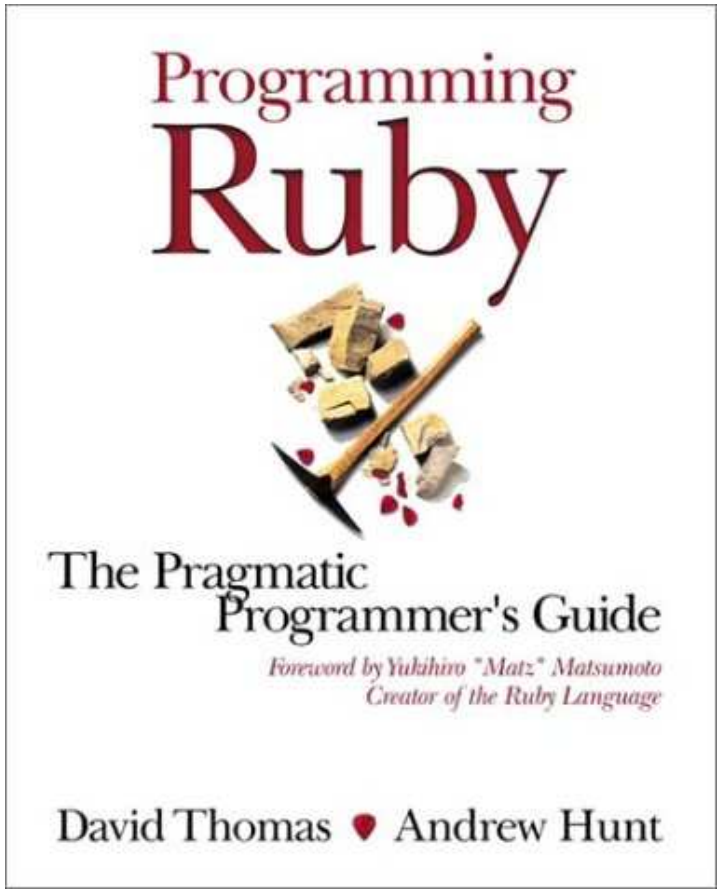
In 1999

First Book



In 2000

First English Book



Ruby in early 2000s

Became a Language
for Geeks

In 2004

Ruby on Rails

Ruby on Rails

Web Application Framework

Web development that doesn’t hurt

Ruby on Rails

- 10x Productive than Java
- 15 Minutes to code Blog

Enterprise Ruby

Ruby started to be used in the Enterprise Environment

Enterprise Ruby

Concerns

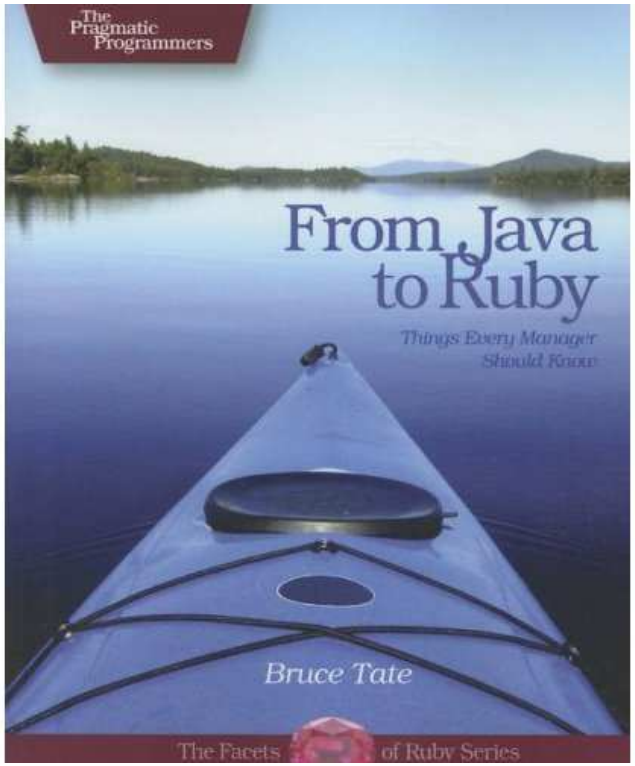
- Fast Enough?
- Scales?

Enterprise Ruby

Issues are Matters of
Resource/Money We Put in.

Ruby's Mindshare

From Java To Ruby



What’s "Enterprise"?

What Major Players Recommend:

- Sun Microsystems
- Microsoft
- Apple
- Etc

Sun Microsystems

JRuby

- Now Pushes Ruby
- Hires Two Core Developers

Microsoft

Silverlight

Dynamic Language Runtime

- IronPython
- IronRuby

Apple

- OSX Shipped with Ruby
- Ruby Cocoa
- XServer Service using Rails
- MacRuby

SAP

- Blue Ruby
  - on their own VM (ABAP)

Oracle

- Oracle Mix
- JRuby on Rails/6 weeks

Twitter

The Biggest Rails Site

Summary

- Moore’s Law Has Changed the World
- We Need Productivity

Summary

- Productivity through Languages
  - Succinctness
  - Good Usability

Summary

- Ruby is Productive
- Ruby is Fun

A Message from Ruby

Enjoy Programming!