



Ruby after 25 years

Heroku
Ruby Association

Yukihiro "Matz" Matsumoto
まつもとゆきひろ
@yukihiro_matz

1993



I started developing Ruby



Just for fun



Total Number of Users: 1



1995



I put it on the Internet

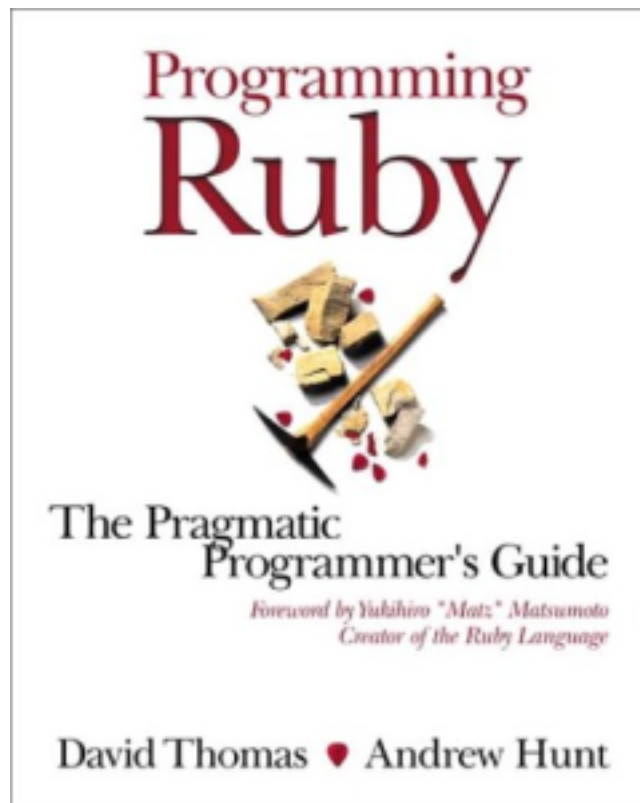


Total Number of Users: 200

2000



The Pickaxe Book





Total Number of Users: 10000?

2001



The first RubyConf



In Tampa, Florida



Attendees: 30+

2017



RubyKaigi in 広島 (Hiroshima)



Attendees: 800+



Total Number of Users: 1M+?



1 to 1M User(s)



Ruby became Popular



Ruby is used everywhere



From Small to Big



Numerous applications written
in Ruby



Ruby Should be Stable



Compatibility Matters



- Ruby1.9
- Python3
- PHP6
- ECMAScript4



Bigger Community



OSS Community is a weird thing



Community



A group of people



Membership



Non-exclusive



Can't expect strong loyalty



We need to attract community



Otherwise community members
will go away



We will eventually fade away



OSS Community is like a shark



We have to keep swimming



OSS cannot stop



Otherwise it will die



I am sick of hostile claims




"Ruby is Dead"



Because Ruby gems have less
GitHub stars on GitHub



Because Ruby doesn't have
static typing



Because Ruby is no longer shiny
language



Any technology cannot stay
shiny for long



Ruby has 20+ years of history



Rails has 10+ years old



More history, more burden



We have to do something to
survive



What shall we do?



Rebuilding the language?



- Perl6
- PHP6
- JavaScript (ES4)
- Python3
- Ruby1.9



Keeping the compatibility
forever?

- 
- FORTRAN
 - COBOL



Contradiction



Designing a Language is Hard



Harder than You May Think



But that is the way



- Keep compatibility
- Keep progressing



Keep compatibility



Keep existing Ruby program
running



Keep Ruby being Ruby



Keep progressing



Make Ruby run faster



Make Ruby more productive

Ruby3



Ruby3 is the Future



We Will Keep Compatibility



But We Will Make It Great



- Faster
- More Productive



Faster Ruby3

Ruby3x3



Ruby3 will be 3x Faster than
Ruby2



JIT



MJIT



2.8+ times faster (than Ruby2.0)



Performance Solves Many Problems



More Productive Ruby3



Static Analysis



Duck Inference



Interactive Typing



Interactive Programming



A lesson learned



Ruby is not the fastest language



Ruby is not the most powerful
language



We use Ruby because Ruby is
good enough



We use Ruby because Ruby is
comfortable



We use Ruby because Ruby is
nice



Human centric design



Ruby is a nice language



Being nice could bring power



Be nice



Compatibility



Compatibility is Important



Not because we are conservative



Not because we are afraid of
changes



But because it makes us progress



Without compatibility,
community keep using old
versions



Don't pursue beauty of the
software



It's a selfish trap for designers



Consider users



Talk with potential users





Our prediction



Ruby needs Performance



No one complains faster Ruby



Ruby3x3



Programming will be more
conversational



Interactive programming



We have a lot of things to be
done



We have to work with other tools



But I am sure it will benefit the
community



We try to be nice



We are trying make Ruby nicer



That means more powerful Ruby



more comfortable Ruby



and faster Ruby



Obviously Ruby community is
not just Ruby



Gems can be nicer



Your apps can be nicer



We need your cooperation



STRONGER TOGETHER



Summary



Be nice



We can be stronger by being
nicer



Being nice includes:

- Moving forward to survive
- Keeping Compatibility
- Gradual changes
- Providing benefits



Moving forward to survive

Ruby3



Our goal



Our policy



Not because it's easy, but
because we need goal



Challenging goal



Compatibility



We will not break compatibility



For no reason



Gradual changes



We won't throw things away and
restarting from scratch



It may take longer time, but it's
worth it



Ruby3 features will be
introduced to 2.x



Keep providing benefits



Bottlenecks may not be in Ruby



Gems/Frameworks/Apps



We need more tools and
contributions



Those principles can be applied to anything:

- Moving forward to survive
- Keeping Compatibility
- Gradual changes
- Providing benefits




That's All Folks!



One more thing...



Ruby25



2018.2.24 Sat.
Shinagawa Intercity Hall, Tokyo



<http://25.ruby.or.jp>



#Ruby25



Thank you