

BOOTSTRAPPING RAILS

A CAREER IN RUBY

1



Matt Valentine-House

Senior Developer, Ruby Infrastructure



@eightbitraptor@ruby.social



@eightbitraptor

2

Ruby & Rails. Past & Present

What does a career in Ruby look like

Developing Ruby

Why Ruby is still a good choice

3

RUBY & RAILS: ORIGINS

4

Object Oriented Scripting Language

Originated in Japan in 1993

Influenced by Lisp, Perl, Smalltalk

First public release (v0.5) in 1995

First English mailing list in 1999





7

**Extracted generic code from Basecamp
ersatz MVC pattern
ActiveRecord
Templating system using ERB
Routing layer to forward requests**



9

2004: Rails & Basecamp

2005: Shopify

2006: Ruby TIOBE Language of the year

2007: Github launched

10



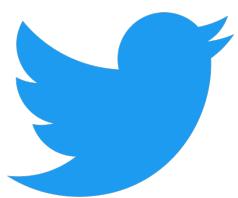
500px

stripe

Bloomberg



goodreads



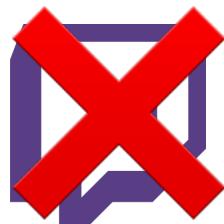
11



500px

stripe

Bloomberg



goodreads



12



13



2005: BBC starts employing Rails developers

2006: Programme Archive opens, built in Rails

14



Government Digital Service

15

GOV.UK

The best place to find government services and information

Search

Popular on GOV.UK

- Get support with the cost of living
- Find out about help you can get with your energy bills
- Find a job
- Universal Credit account: sign in
- Check your National Insurance record
- Check MOT history of a vehicle

Services and information

Benefits

Includes eligibility, appeals, tax credits and Universal Credit

Featured

Cost of Living Payment

Find out payment dates and how to report a

16

Initial team of 16 people

Launched a beta in ~3 months

Built in Ruby

Developed in the open on Github

Code is MIT licensed

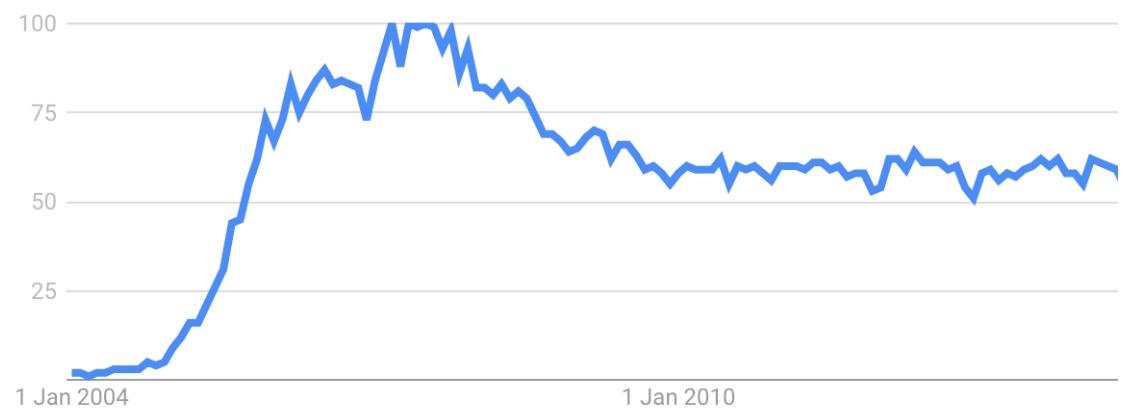
Open Government License (CC v4.0)

17

The screenshot shows the Canadian Digital Service website. The top navigation bar includes the Canadian flag, "Government of Canada / Gouvernement du Canada", a search bar, and links for "Français", "Products", "About", and "Blog". The main heading is "Digital products for the Government of Canada" built by the Canadian Digital Service. Below this is a large illustration of four people working together on a large document or interface. Two product cards are displayed at the bottom:

- GC Notify**: "Keep the public informed every step of the way. Try GC Notify, a better way to send service updates to your clients." It includes "Create an account" and "Sign in" buttons.
- GC Forms**: "Publish government forms online. Make it easier for people to submit information to your service by building simple, accessible, online forms — no coding required." It includes "Get started", "Contact us", and "Sign in" buttons.

18



Google Trends: Ruby on Rails

19

```
class HelloWorldRackApp
  def call(env)
    [200,
     {"content-type" =>
      "text/html; charset=utf-8"},,
     ["Hello World"]]
  end
end

run HelloWorldRackApp.new
```

20

```
class HelloWorldRackApp
  def call(env)
    [200,
     {"content-type" =>
      "text/html; charset=utf-8"},,
     ["Hello World"]]
  end
end

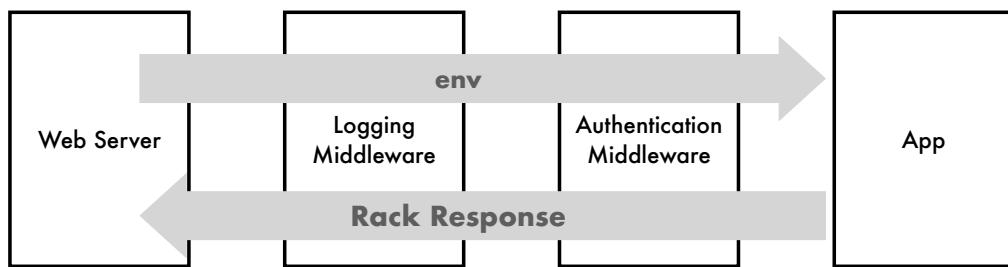
run HelloWorldRackApp.new
```

21

```
class HelloWorldRackApp
  def call(env)
    [200,
     {"content-type" =>
      "text/html; charset=utf-8"},,
     ["Hello World"]]
  end
end

run HelloWorldRackApp.new
```

22



23

```
class HelloWorldRackApp
  def call(env)
    [200,
     {"content-type" =>
      "text/html; charset=utf-8"},,
     ["Hello World"]]
  end
end

run HelloWorldRackApp.new
```

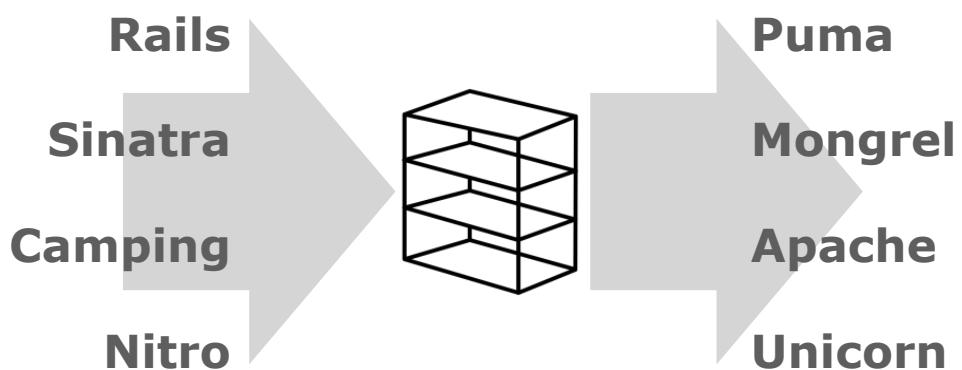
24

```
require 'sinatra'

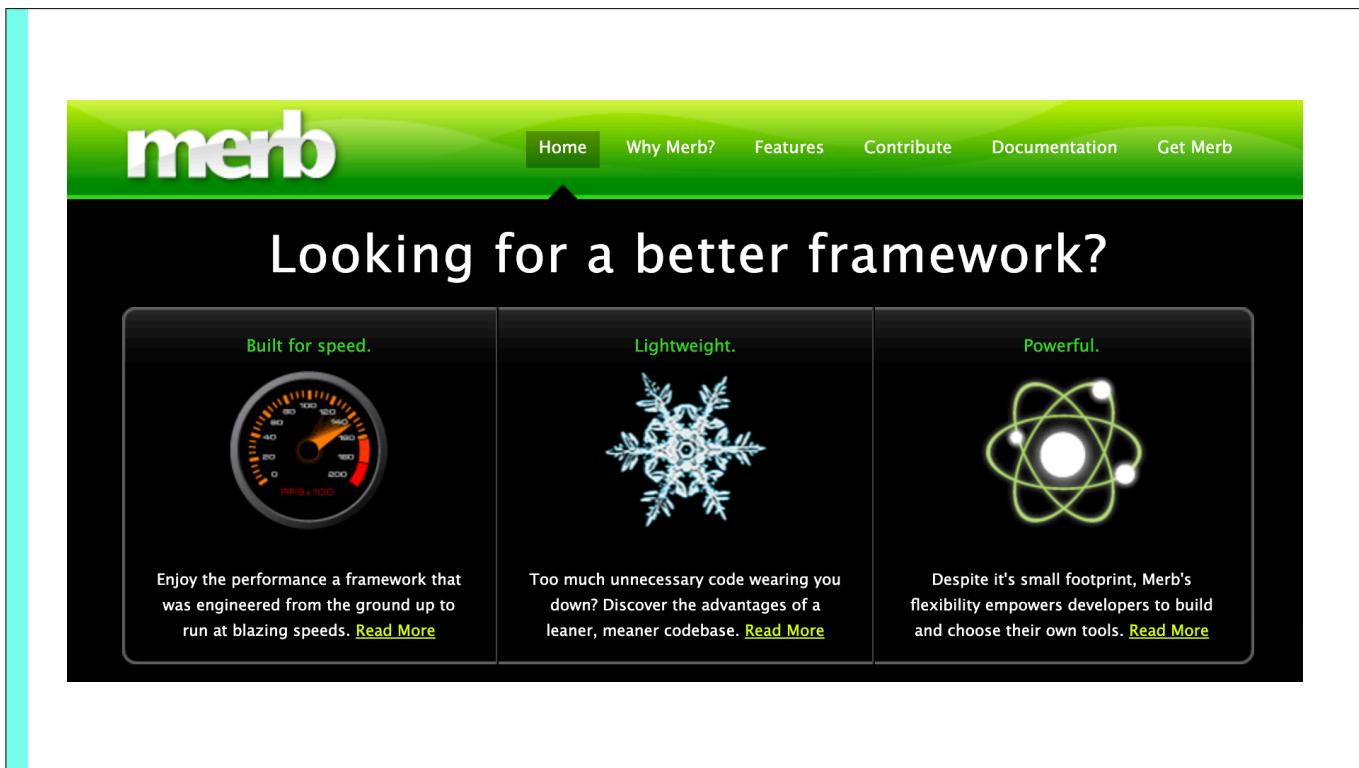
class HelloWorldRackApp < Sinatra::Application
  get '/' do
    headers {"content-type" =>
              "text/html; charset=utf-8"}
    body "Hello World from Sinatra"
  end
end

run HelloWorldRackApp.new
```

25



26



The screenshot shows the Merb homepage. At the top is a green header bar with the Merb logo on the left and navigation links: Home, Why Merb?, Features, Contribute, Documentation, and Get Merb. Below the header is a large black section containing the text "Looking for a better framework?". Underneath this are three cards: "Built for speed." with a tachometer icon, "Lightweight." with a snowflake icon, and "Powerful." with an atom icon. Each card has a brief description and a "Read More" link.

Built for speed.



Enjoy the performance a framework that was engineered from the ground up to run at blazing speeds. [Read More](#)

Lightweight.



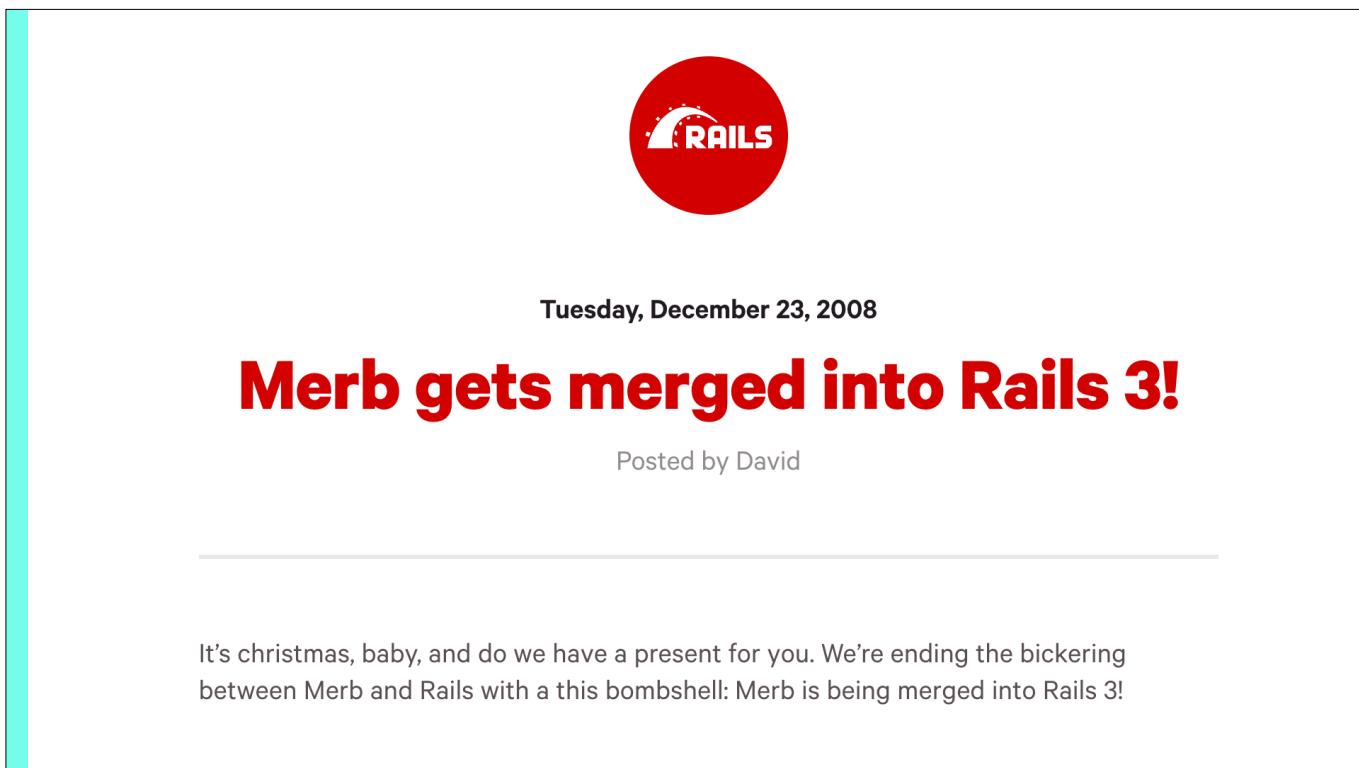
Too much unnecessary code wearing you down? Discover the advantages of a leaner, meaner codebase. [Read More](#)

Powerful.



Despite it's small footprint, Merb's flexibility empowers developers to build and choose their own tools. [Read More](#)

27



The screenshot shows a blog post titled "Merb gets merged into Rails 3!" by David on Tuesday, December 23, 2008. The post begins with a statement about Merb being merged into Rails 3.

Tuesday, December 23, 2008

Merb gets merged into Rails 3!

Posted by David

It's christmas, baby, and do we have a present for you. We're ending the bickering between Merb and Rails with a this bombshell: Merb is being merged into Rails 3!

28



29



Tuesday, December 23, 2008

Merb gets merged into Rails 3!

Posted by David

It's christmas, baby, and do we have a present for you. We're ending the bickering between Merb and Rails with a this bombshell: Merb is being merged into Rails 3!

30

Myriad local communities worldwide

Rails Girls

WNB.rb

31

A CAREER IN RUBY

32



33



34

2007-2011

Systems Administrator for 1.5 years

Introduced DevOps

Transitioned to a Full-time Developer

Ruby 1.8 -> 1.9, and Rails 2 -> 3

Got involved with LRUG

35



36

2011-2013

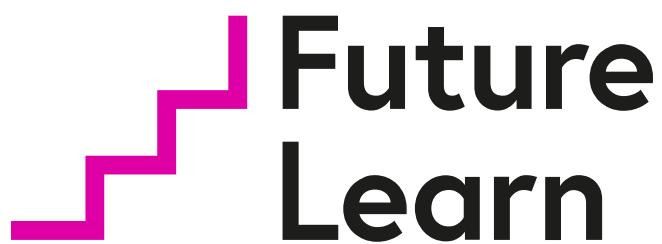
Much bigger team than Reevoo

Larger, more complex product

Rewriting from Perl into Ruby & Rails

Regular at NWRUG & Leeds Ruby Thing

37



38

2013-2018

Full time developer #6

Greenfield Rails application

Datawarehouse & ETL entirely in Ruby

Business Analytics for company of >250

39



40

Milestones

SHOPIFY PLATFORM RELEASED
2006

EMPLOYEES (AS OF 12/31/23)
8,300+

MERCHANTS
millions of merchants

TOTAL SALES ON SHOPIFY
~\$886 Billion

MILLIONS OF MERCHANTS IN
~175 Countries

APPS IN OUR APP STORE
13,000+

THEMES IN OUR THEME STORE
70

OF EXPERTS IN OUR NETWORK
780+

41



Tobi Lütke, Founder & Chief Executive Officer

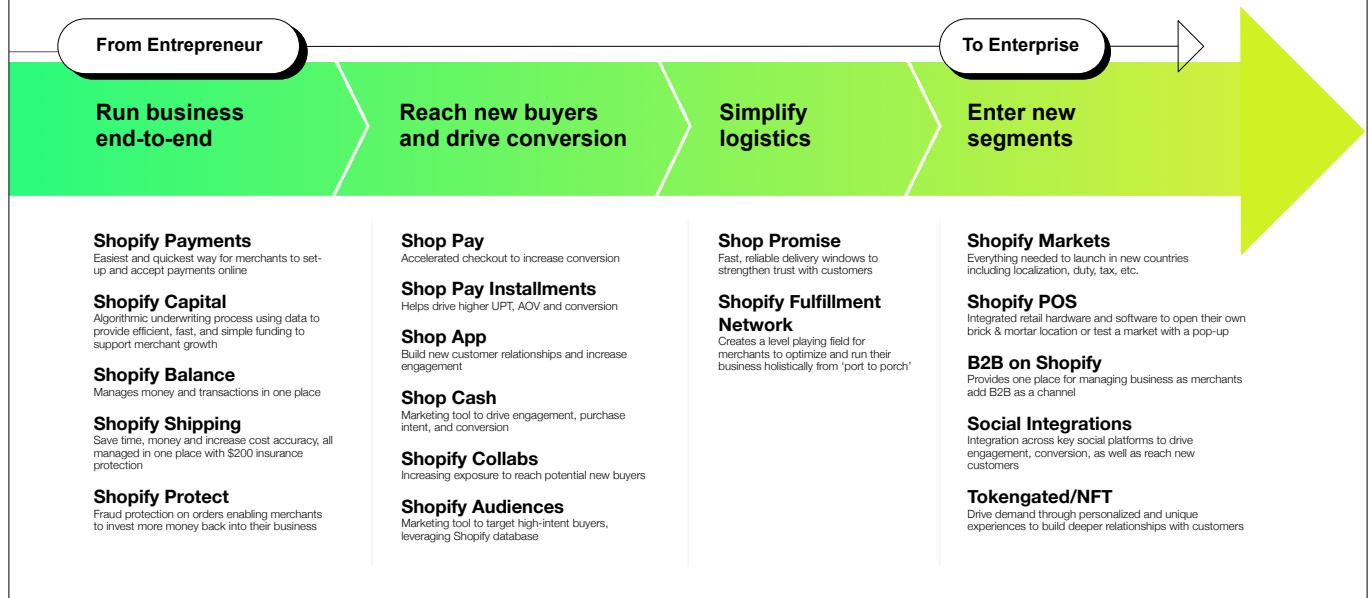
Tobi Lütke is the founder and Chief Executive Officer of Shopify. In 2004, Tobi began building software to launch an online snowboard store called Snowdevil. It quickly became obvious that the software was more valuable than the snowboards, so Tobi and his founding team launched the Shopify platform in 2006. He has served as CEO since 2008 at the company's headquarters in Ottawa, Canada.

Tobi is an active advocate for sustainability, computer literacy, and education, and serves as a board member of Canada Learning Code, an organization working to give all Canadians access to digital skills. He also served as Chair of the Digital Industries Table, an advisory board commissioned by the federal government to provide recommendations on how to turn Canada into a digital leader.

42

Building mission critical solutions to **run a merchant's entire business at every stage**

Leverage software development expertise and economies of scale to advance commerce



43

Consumers' favorite brands powered by Shopify



44

The screenshot shows the homepage of the Shopify Rebellion website. At the top, there's a navigation bar with links for 'STORE', 'OUR REBELS', and 'BLOG'. A live streaming indicator for 'Zizaran' with 6639 viewers is also present. Below the navigation, several game logos are displayed: DOTA 2, HALO, ROCKET LEAGUE, STAR CRAFT, VALORANT, and LEAGUE OF LEGENDS. The main content area features a portrait of a man named Dario Wünsch, identified as the 'HEAD OF PLAYER DEVELOPMENT'. A quote from him discusses the values of esports, mentioning 'Fair play, respectin...'. A 'READ MORE' button is located at the bottom right of this section.

45

Engineering at Shopify

Predominantly powered by Ruby. The main core monolith has > **2.8 million Ruby loc**, spread over **40,000 Ruby files**, not including tests (c. 2020). Core also has >**230,000 automated tests**. It's also **18 years old**

1300 Pull Requests per day, **1000 deploys** per day. On average **40 new Shopify versions per day**.

Average **5 million requests per minute** (83k per second). Flash sales have caused peaks of **2.5 million requests a minute peak**, and **>1 million unique visitors in 60 minutes** for a single shop.

46

\$9.3 billion USD of sales

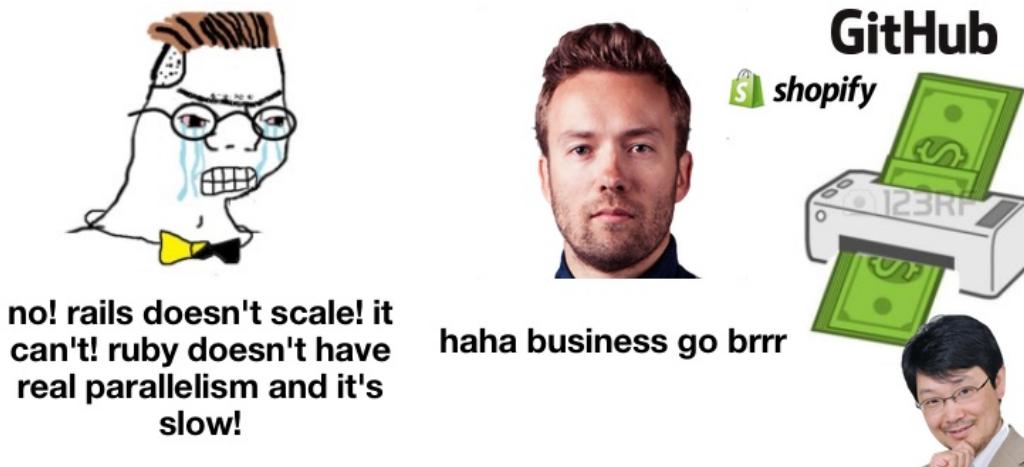
\$4.1 million dollars/minute peak

Greater than 99.999% uptime

58 million requests/minute peak

29.7 Pb of data served from our infra

47



48

EVOLUTION OF RUBY

49

CRuby/MRI & YARV

Alternative Ruby implementations

Garbage Collection

JIT Compilation

Parsing Ruby

50

Ruby originally a tree-walking interpreter

YARV started in 2004 by Sasada Koichi

Stack based virtual machine

Merged into CRuby in 2007 - 1.9.0

Huge speed improvements vs MRI

```
def say_hello(name)
  puts "#{name} こんにちは"
end

say_hello("みなさん")
```

```
ruby --dump=insnns test.rb
```

53

```
== disasm: #<ISeq:<main>@test.rb:1 (1,0)-(5,25)>
0000 definemethod                      :say_hello, say_hello      ( 1)[Li]
0003 putself                           ( 5)[Li]
0004 putstring                         "みなさん"
0006 opt_send_without_block            <calldata!mid:say_hello, argc:1, FCALL|ARGS_SIMPLE>
0008 leave

== disasm: #<ISeq:say_hello@test.rb:1 (1,0)-(3,3)>
local table (size: 1, argc: 1 [opts: 0, rest: -1, post: 0, block: -1, kw: -1@-1, kwrest: -1])
[ 1] name@0<Arg>
0000 putself                           ( 2)[LiCa]
0001 putobject                         "こんにちは"
0003 getlocal_WC_0                     name@0
0005 dup
0006 objtostring                      <calldata!mid:to_s, argc:0, FCALL|ARGS_SIMPLE>
0008 anytostring
0009 concatstrings                    2
0011 opt_send_without_block           <calldata!mid:puts, argc:1, FCALL|ARGS_SIMPLE>
0013 leave                            ( 3)[Re]
```

54

Ruby originally a tree-walking interpreter

YARV started in 2004 by Sasada Koichi

Stack based virtual machine

Merged into CRuby in 2007 - 1.9.0

Huge speed improvements vs MRI

Bytecode Introspection & Profiling

Inline method caches

Compile time optimisations

Specialised instructions

Native thread

```
puts RubyVM::InstructionSequence
  .compile_file("test.rb")
  .disasm
```

57

```
== disasm: #<ISeq:<main>@test.rb:1 (1,0)-(5,25)>
0000 definemethod :say_hello, say_hello      ( 1)[Li]
0003 putself          ( 5)[Li]
0004 putstring         "みなさん"
0006 opt_send_without_block <calldata!mid:say_hello, argc:1, FCALL|ARGS_SIMPLE>
0008 leave

== disasm: #<ISeq:say_hello@test.rb:1 (1,0)-(3,3)>
local table (size: 1, argc: 1 [opts: 0, rest: -1, post: 0, block: -1, kw: -1@-1, kwrest: -1])
[ 1] name@0<Arg>
0000 putself          ( 2)[LiCa]
0001 putobject         "こんにちは"
0003 getlocal_WC_0    name@0
0005 dup
0006 objtostring       <calldata!mid:to_s, argc:0, FCALL|ARGS_SIMPLE>
0008 anytostring
0009 concatstrings     2
0011 opt_send_without_block <calldata!mid:puts, argc:1, FCALL|ARGS_SIMPLE>
0013 leave             ( 3)[Re]
```

58

Bytecode Introspection & Profiling

Inline method caches

Peephole optimisations

Specialised instructions

Native thread

59

CRuby/MRI & YARV

Alternative Ruby implementations

Garbage Collection

JIT Compilation

Parsing Ruby

60

Ruby Enterprise Edition

IronRuby (.NET CLR)

MacRuby (Obj-C Runtime)

jsruby (node.js)

RedSun (Flash)

61

JRuby (JVM)

mRuby (Embedded)

TruffleRuby (GraalVM)

62

JRuby (JVM)

mRuby (Embedded)

TruffleRuby (GraalVM)

63



64

JRuby (JVM)

mRuby (Embedded)

TruffleRuby (GraalVM)

65

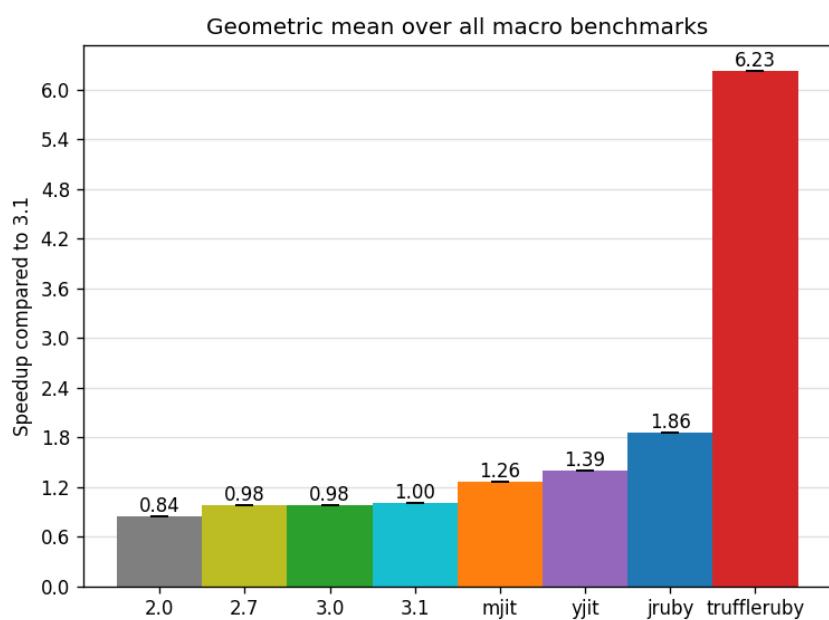


Truffle AST interpreter framework

Graal VM optimised JIT

Sponsored by Shopify since 2019

66



CRuby/MRI & YARV

Alternative Ruby implementations

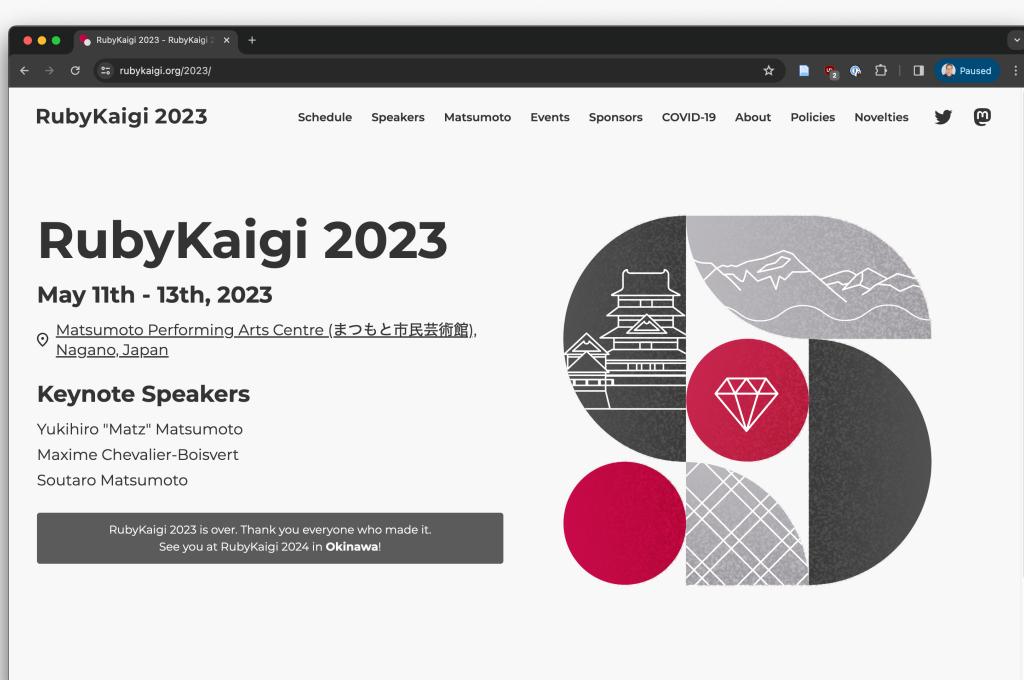
Garbage Collection

JIT Compilation

Parsing Ruby

562 → 14100

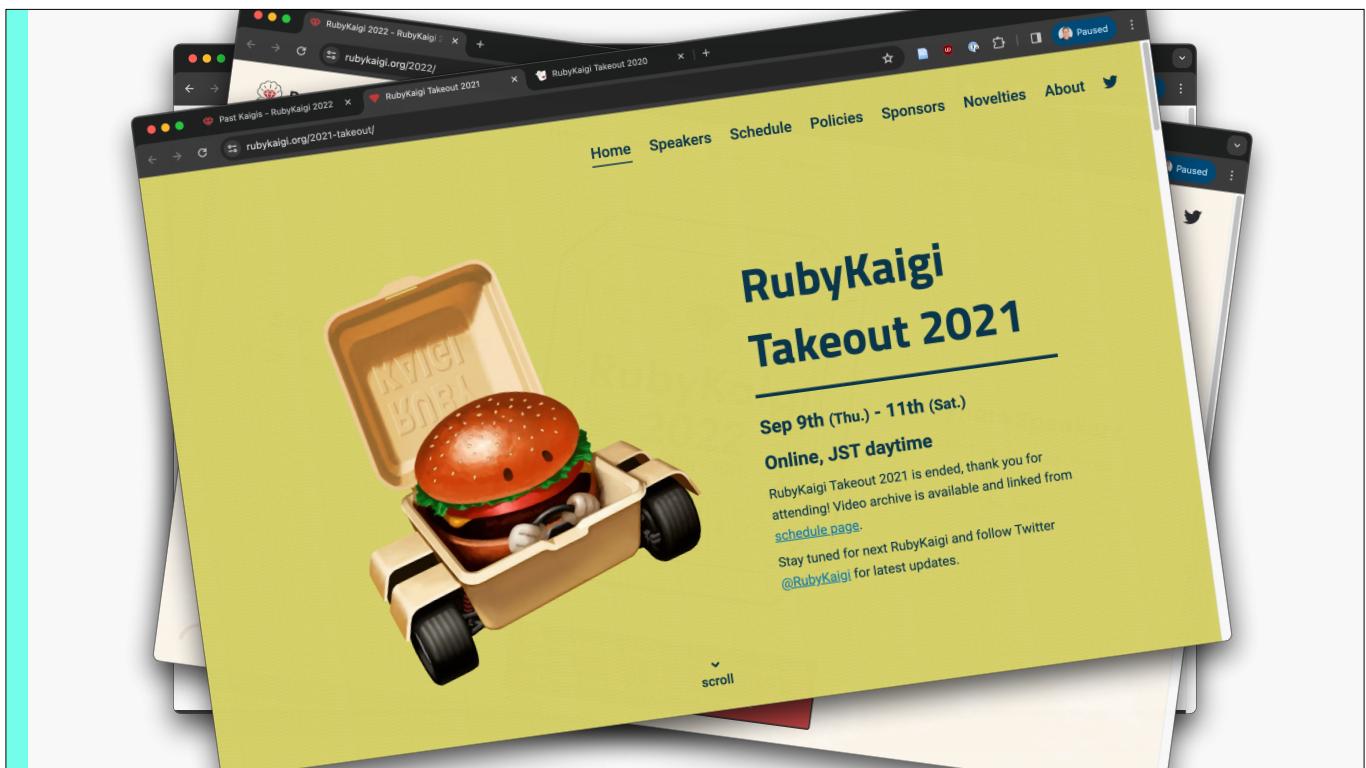
69



70



71



72

“Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part 1”

*John McCarthy
Massachusetts Institute of Technology*

Communications of the ACM, Volume 3, Issue 4, April 1960

73

2011 (1.9.3): Lazy Sweeping

2013 (2.1.0): Generational GC

2014 (2.2.0): Incremental Marking

2019 (2.7.0): GC Compaction

2020 (3.0.0): Automatic Compaction

2022 (3.2.0): Variable Width Allocation

74



75

- Provide a modular GC Interface**
- Experimentation with alternative GC's**
- Increase the academic viability of Ruby**
- Easy split-testing of GC tunings at Runtime**

76

CRuby/MRI & YARV

Alternative Ruby implementations

Garbage Collection

JIT Compilation

Parsing Ruby

77

JIT story starts in 2016 with MJIT

Method based JIT

YARV Instructions inlined into C

MJIT actually lowered Rails performance

78

In 2020 YJIT development began
Lazy Basic Block Versioning
20% improvement within a year
Released with Ruby 3.1 in 2021

79

In 2022 YJIT was rewritten in Rust
Memory usage cut by 66%
Performance continued to improve
YJIT-Bench built to track performance

80

YJIT Benchmarks

How is YJIT's speed on its [benchmarks](#) as of 2024-02-20 19:08:13?

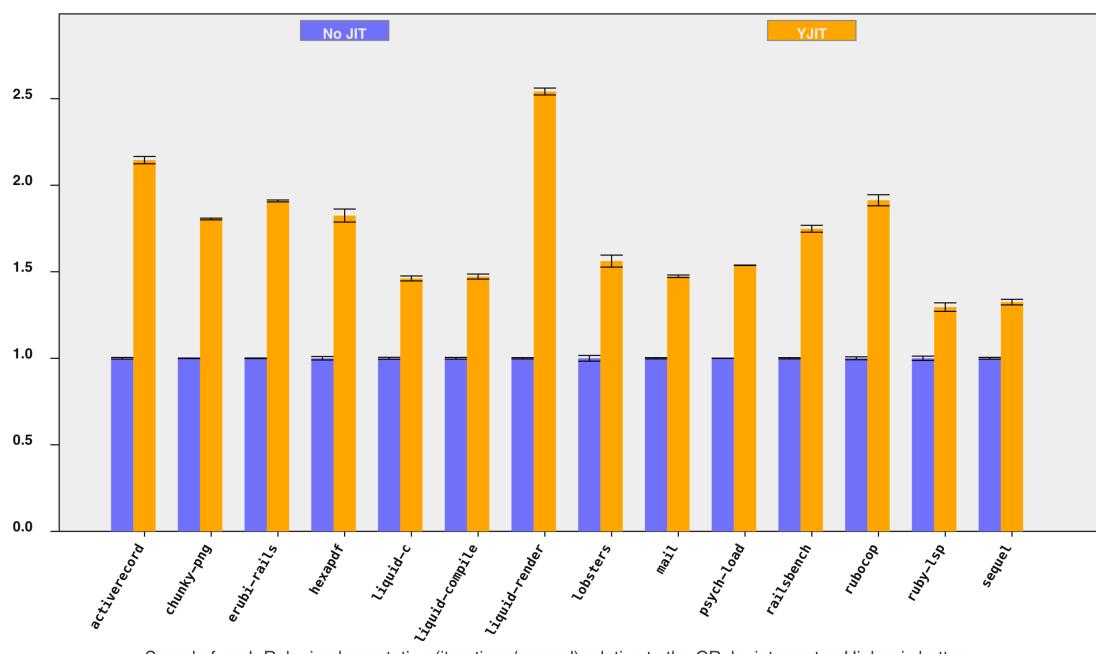
Overall YJIT is 68.6% faster than interpreted CRuby!
On Railsbench specifically, YJIT is 74.9% faster than CRuby!

[Latest Full Details \(x86_64\)](#)

[Latest Full Details \(aarch64\)](#)

<https://speed.yjit.org>

81



82

DHH · @dhh · ...

YJIT is going to be enabled by default in the next version of Rails when running on Ruby 3.3+. We saw an incredible 10-24% increase in performance on Basecamp and HEY after switching to YJIT. Ruby is so fast these days, it's great 🎉

rails/rails

Enable YJIT by default if running Ruby 3.3+

+31 -0 ████

byroot committed November 7, 2023 · c29cda2

Enable YJIT by default if running Ruby 3.3+ · rails/rails@c29cda2

From github.com

4:35 PM · Nov 9, 2023 · 90.3K Views

83

CRuby/MRI & YARV

Alternative Ruby implementations

Garbage Collection

JIT Compilation

Parsing Ruby

84

Maintainability

Error Tolerance

Portability

Performance

85

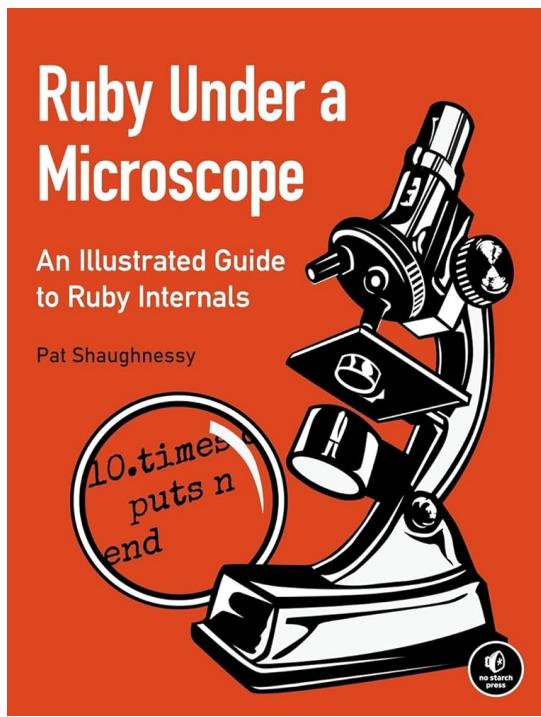
Maintainability

Error Tolerance

Portability

Performance

86



87

```
irb(main):001> def foo = puts("bar")
```

88

```
irb(main):001> def foo = puts("bar")
=> :foo
irb(main):002> foo
bar
=> nil
```

89

```
irb(main):003> puts("bar") and puts("baz")
```

90

```
irb(main):003> puts("bar") and puts("baz")
bar
=> nil
```

91

```
irb(main):006> def foo = puts("bar") and puts("baz")
```

92

```
irb(main):001> def foo = puts("bar") and puts("baz")
baz
=> nil
irb(main):002> foo
bar
=> nil
```

93

```
irb(main):001> (def foo = puts("bar")) and puts("baz")
```

94

Home My page Projects Help

Logged in as **eightbitraptor** My account Sign out

Ruby » Ruby master

Search: Ruby master

Overview Activity Roadmap Issues New issue Repository Mail to issue Like Settings

Bug #19392 CLOSED

 Endless method and parsing priorities
Added by zverok (Victor Shepelev) about 1 year ago. Updated 2 months ago.

Status: Closed
Priority: Normal
Assignee: -
Target version: -
ruby -v: **Backport:** 2.7: DONTNEED, 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN

[ruby-core:112119]

Description Quote

Initial description

Discovered by Lucian Ghinda:

```
def test = puts("foo") and puts("bar")
# prints "bar" immediately
test
# prints "foo"
```

It seems that it is a parser error, right?..

My custom queries

- Inactive Bugs
- My closed tickets
- open bugs
- Tumbleweed bugs

Custom queries

- Backport 3.0
- Backport 3.1
- Backport 3.2
- Backport 3.3
- bugs: unassigned
- DevMeeting
- matz
- Open issues with attachment

Watchers (3) Add

-  Dan0042 (Daniel DeLorme) 
-  kddnewton (Kevin Newton) 
-  bughit (bug hit) 

95

Home My page Projects Help

Logged in as **eightbitraptor** My account Sign out

Ruby » Ruby master

Search: Ruby master

Overview Activity Roadmap Issues New issue Repository Mail to issue Like Settings

Bug #19392 CLOSED

 Endless method and parsing priorities
Added by zverok (Victor Shepelev) about 1 year ago. Updated 2 months ago.

Status: Closed Normal

Priority: Normal

Assignee: -

Target version: -

ruby -v: **Backport:** 2.7: DONTNEED, 3.0: UNKNOWN, 3.1: UNKNOWN, 3.2: UNKNOWN

[ruby-core:112119]

Description Quote

Initial description

Discovered by Lucian Ghinda:

```
def test = puts("foo") and puts("bar")
# prints "bar" immediately
test
# prints "foo"
```

It seems that it is a parser error, right?..

My custom queries

- Inactive Bugs
- My closed tickets
- open bugs
- Tumbleweed bugs

Custom queries

- Backport 3.0
- Backport 3.1
- Backport 3.2
- Backport 3.3
- bugs: unassigned
- DevMeeting
- matz
- Open issues with attachment

Watchers (3) Add

-  Dan0042 (Daniel DeLorme) 
-  kddnewton (Kevin Newton) 
-  bughit (bug hit) 

96

Maintainability

Error Tolerance

Portability

Performance

97

Maintainability

Error Tolerance

Portability

Performance

98

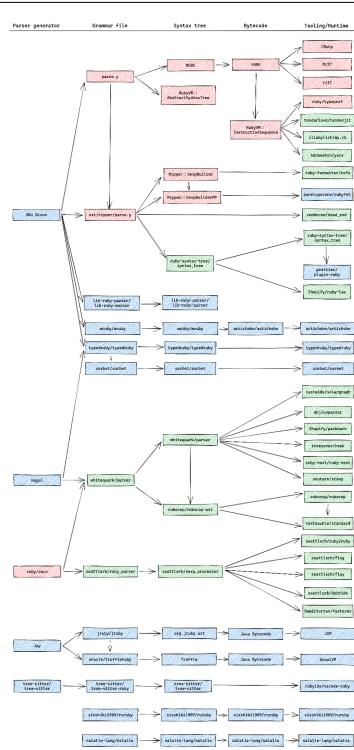
Maintainability

Error Tolerance

Portability

Performance

99



100

CRuby	Ripper
mruby	parser
JRuby	ruby_parser
TruffleRuby	tree-sitter-ruby
Natalie	Sorbet
ruruby	lib-ruby-parser

101

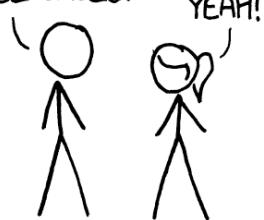
Maintainability
Error Tolerance
Portability
Performance

102

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION:
THERE ARE
14 COMPETING
STANDARDS.

14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.



SOON:
SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

103



104

Hand written Recursive Descent Parser

Portable, Maintainable, and Performant

Standardised Syntax Tree

Already in widespread use

Merged into CRuby for 3.3 in 2023

Hand written Recursive Descent Parser

Portable, Maintainable, and Performant

Standardised Syntax Tree

Already in widespread use

Merged into CRuby for 3.3 in 2023

Hand written Recursive Descent Parser
Portable, Maintainable, and Performant
Standardised Syntax Tree
Already in widespread use
Merged into CRuby for 3.3 in 2023

107

Hand written Recursive Descent Parser
Portable, Maintainable, and Performant
Standardised Syntax Tree
Already in widespread use
Merged into CRuby for 3.3 in 2023

108

CRuby	Ripper
mruby	parser
JRuby	ruby_parser
TruffleRuby	tree-sitter-ruby
Natalie	Sorbet
ruruby	lib-ruby-parser

109

CRuby	✓ Ripper
mruby	✓ parser
✓ JRuby	✓ ruby_parser
✓ TruffleRuby	tree-sitter-ruby
✓ Natalie	✓ Sorbet
ruruby	lib-ruby-parser

110

Hand written Recursive Descent Parser
Portable, Maintainable, and Performant
Standardised Syntax Tree
Already in widespread use
Merged into CRuby for 3.3 in 2023

111

WHY RUBY IN 2024 AND BEYOND

112

Perl

Python

Javascript

113

Perl

Python

Javascript

114

Perl

Python

Javascript

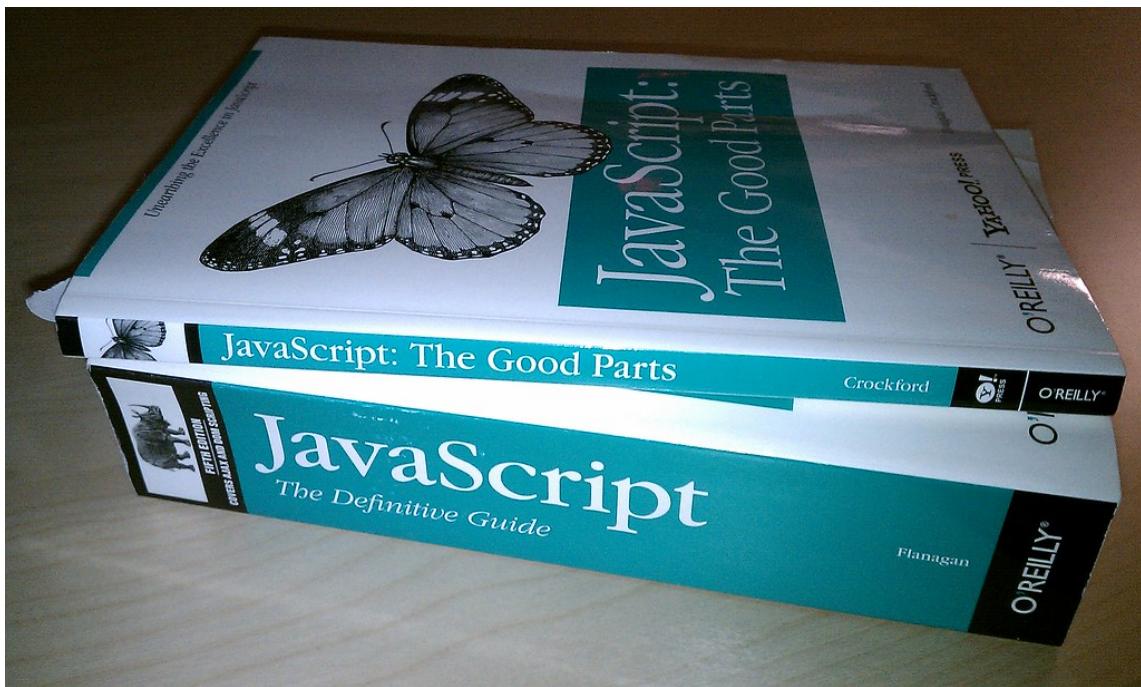
115

Perl

Python

Javascript

116



117

Don't Repeat Yourself
Convention over Configuration
Separation of concerns

118

Rails was revolutionary

119



120

“

Techologies don't stop scaling. But some teams' talent to scale technologies can run out prematurely

”

– Tobi Lutke
Twitter, Nov 25 2023

121

Large, Mature, Friendly Community

Active & engaged core team

Constant language innovation

Huge number of production deployments

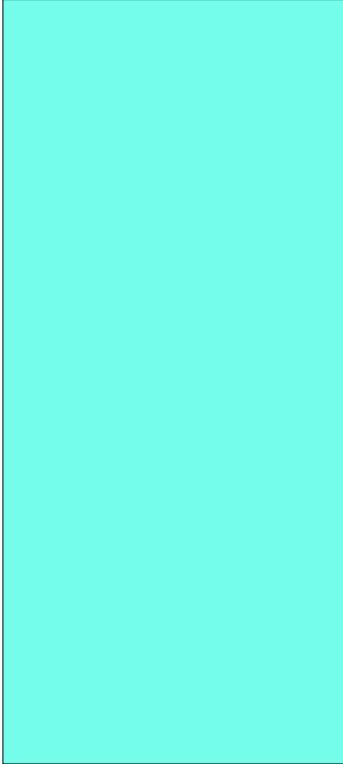
122

Language	Median Salary	Number of Jobs
Swift	\$116,000	30
Python	\$105,000	227
Golang	\$100,000	412
C++	\$100,000	61
Java	\$95,000	134
Ruby	\$95,000	474
Rust	\$88,000	13
SQL	\$85,000	147

123

THANK YOU

124



QUESTIONS?