

A detailed illustration of a futuristic city at night. The scene is viewed from an elevated position, looking down onto a city filled with tall, dark buildings. Many windows and architectural details are illuminated with vibrant blue and purple light. Several vertical red laser beams or light pillars cut through the dark sky. In the foreground, a modern, dark-colored train platform with glass railings is visible. A small, sleek train is stopped at the platform, and a single figure stands near it. The overall atmosphere is one of a high-tech, cyberpunk-inspired urban environment.

Ruby and Rails

First impressions

```
$ cat WhoAmI.txt
```

```
> Calebe Oliveira
```

📁 Software Engineer @ Paggue

♥ Ruby, Nodejs, PHP

👉 <https://calebe.xyz>



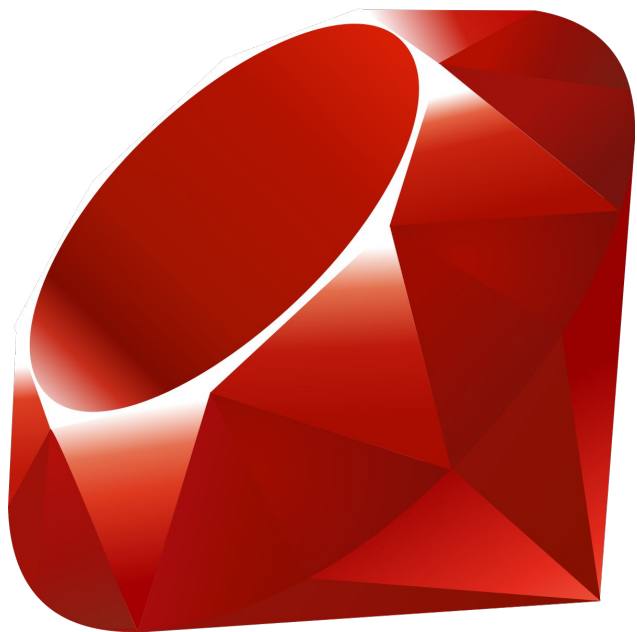
Farol da /
SSA

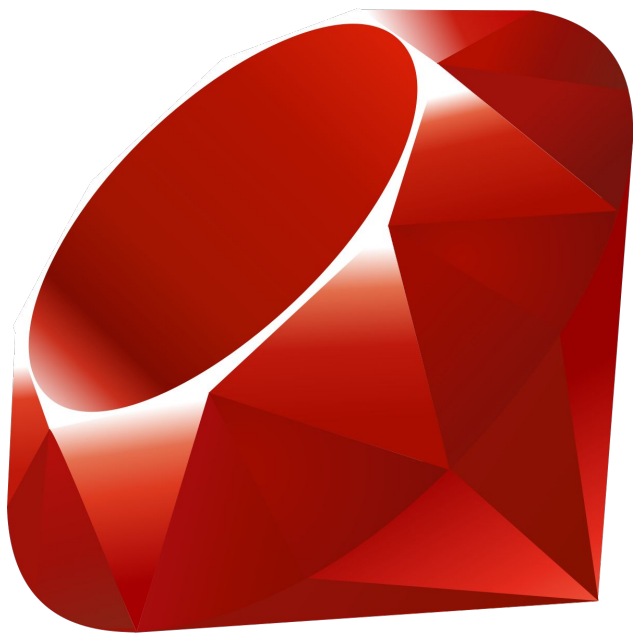
~ 4 months





paggle





!=





!=







Banda

DJ JAVU

e **DJ JUNINHO PORTUGAL**



Yukihiro Matsumoto



Conceived in 1993

First released in 1995 (~23 yo)

Scripting language

“Human oriented”



Caught my eyes

- Dynamic
- Reflective
- Fully object-oriented
- Human friendly syntax

"As a language maniac and OO fan for 15 years, I really wanted a genuine object-oriented, easy-to-use scripting language. I looked for but couldn't find one. So I decided to make [Ruby]."

Matz, 1999

DSL

Domain specific languages

```
Rails.application.routes.draw do
  root to: "pages#main"

  resources :posts do
    get :preview

    resources :comments, only: [:new, :create, :destroy]
  end
end
```



Brainf#ck-ish pitfalls

- Homo C-piens *not so* friendly
- Performance (CRuby)
- Catch Exception and freeze the program (wtf?!)

(Finite) list: <https://github.com/sinelaw/ruby-pitfalls>


```
class A
  @@context = "class"

  def initialize
    @context = "instance"
  end

  def context
    @context
  end

  def A.context
    @@context
  end
end
```

```
a = A.new
a.context # >> "instance"
A.context # >> "class"
```

```
a = [1, 'hi', 3.14, 1, 2, [4, 5]]
```

```
a[2]                # => 3.14
```

```
a.reverse           # => [[4, 5], 2, 1, 3.14, 'hi', 1]
```

```
a.flatten.uniq      # => [1, 'hi', 3.14, 2, 4, 5]
```

```
a.push 23           # a = [1, 'hi', 3.14, 1, 2, [4, 5], 23]
```

```
a << 22             # a = [1, 'hi', 3.14, 1, 2, [4, 5], 23, 22]
```

!-ended method (usually "dangerous")

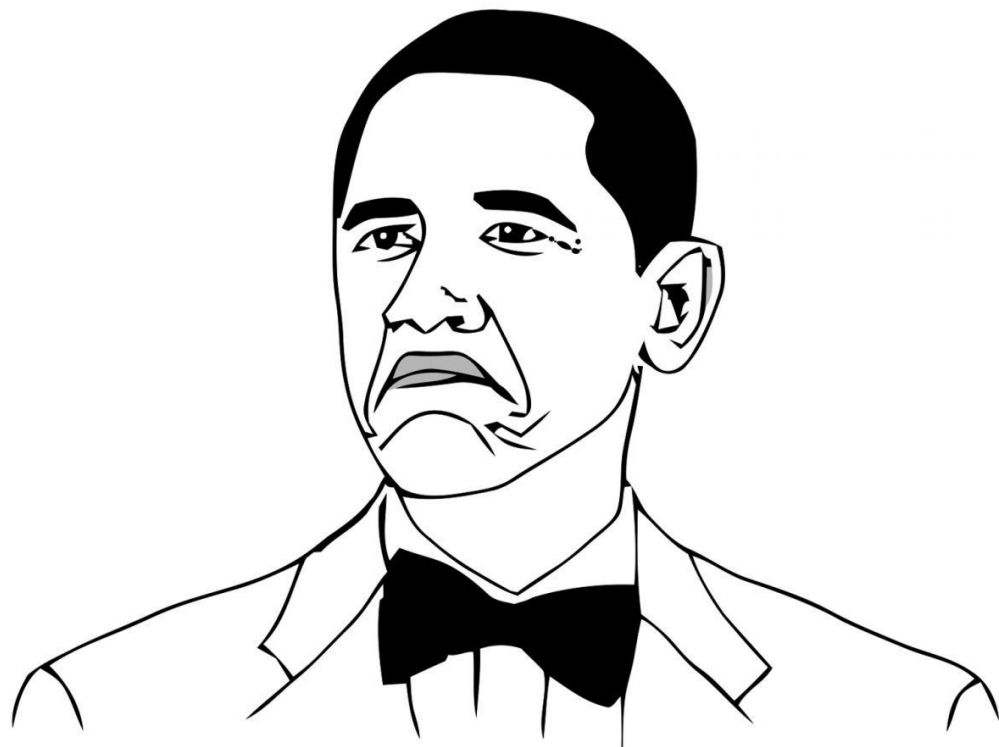
```
b = [1, 2, 3]
```

```
b.reverse!          # b = [3, 2, 1]
```

```
b                   # => [3, 2, 1]
```

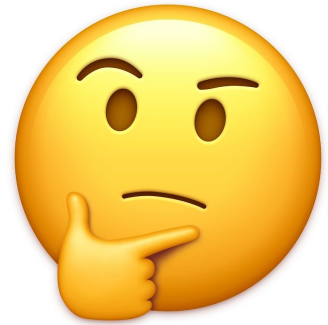
?-ended method (usually bool return)

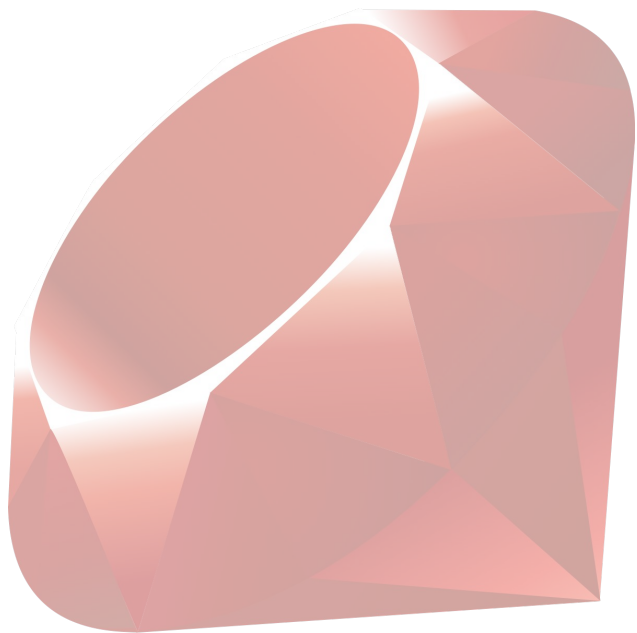
```
nil.nil?            # => true
```



NOT BAD

Devs *shall not* live by beautiful
syntax alone





David Hansson



First released in 2005

MVC-ish, CoC, DRY

Ruby's Django-like

Full-stack meta-framework



Built in frameworks

- Active Record
 - ◆ ORM
- Action Pack
 - ◆ Action View
 - HTML, CSS, Js
 - ◆ Action Controller
 - Business logic control
- Action Mailer
 - ◆ Send and receive emails
- Active Support
 - ◆ “Utils” stuff



Fully grown full-stack

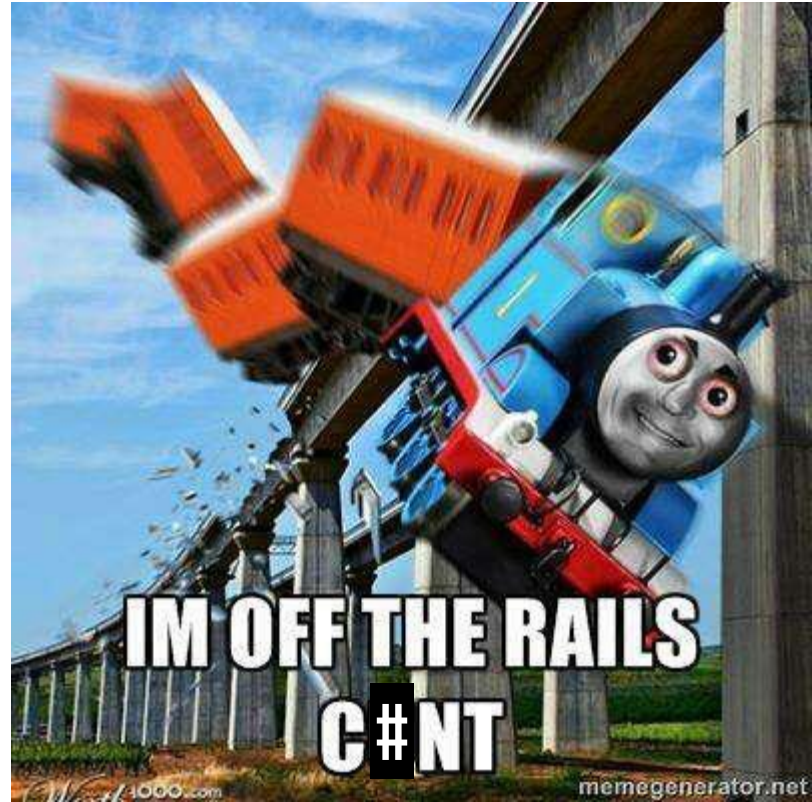
```
$ rails g
```

gem 'devise'

Built-in form builder

Use the goddam conventions

(for God's sake!)



CHOO
CHOO



MOTHERF#CKER

**BUILD IT WITH RAILS, THEY
SAID**



**YOUR DATA WILL BE SAFE,
THEY SAID**

Performance

Verdict

- Ruby and Rails kept my dev flame alive
- Some younger full stack frameworks got some ideas from Rails
- Active Record still sucks sometimes (BBOM)
- Do not fight the framework, you'll fail
- Very useful and practical for Web
- Plenty of job opportunities

Keep on Rails! Thank you!

