



HACKER SCHOOL
MONTERREY

Rails101

Mayra Cabrera

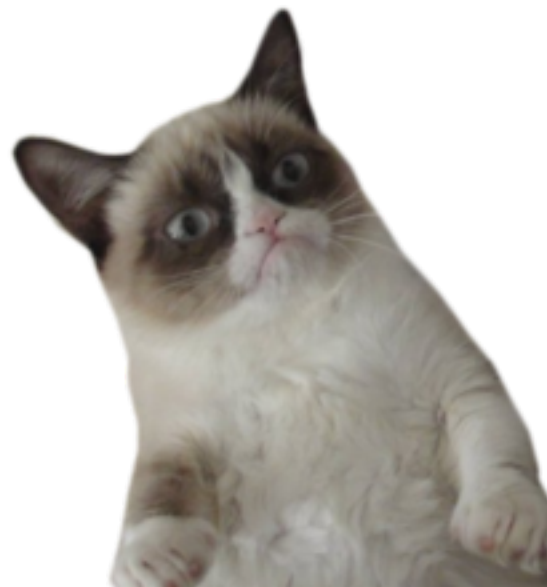
About me

Software engineer at Icalia Labs

I ❤️ Programming

I ❤️ Rails

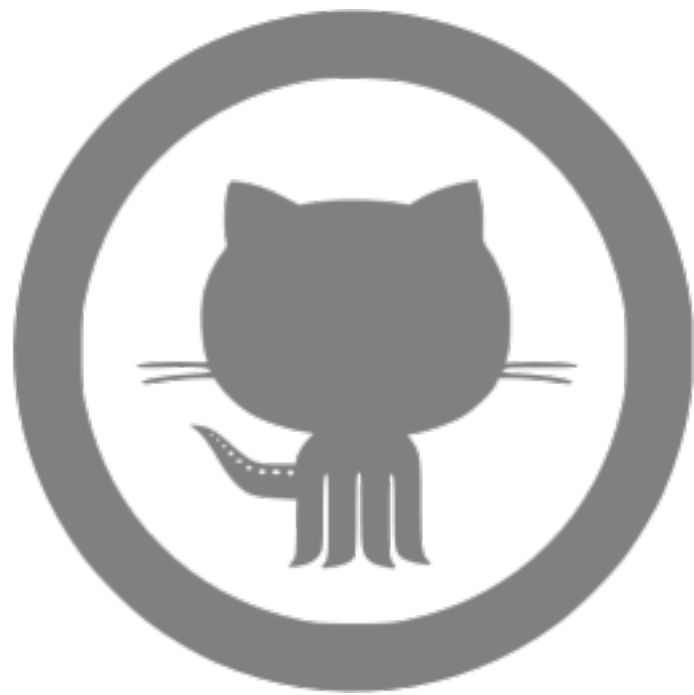
Series, movies, books & cats



About you?



About this class



<https://github.com/hackerschoolmty/Rails101>

Topics

1. Introduction to Rails

- Why Rails
- Philosophy
- MVC architecture
- RESTful architecture

2. Getting started with Rails

- Basic commands
- Structure of rails app
- ActiveRecord
- Controllers
- Views

3. Basic routing

4. User management

5. Debugging

Topics

1. Introduction to Rails

- Why Rails
- Philosophy
- MVC architecture
- RESTful architecture

2. Getting started with Rails

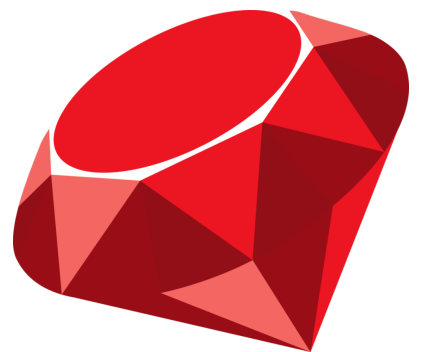
- Basic commands
- Structure of rails app
- ActiveRecord
- Controllers
- Views

3. Basic routing

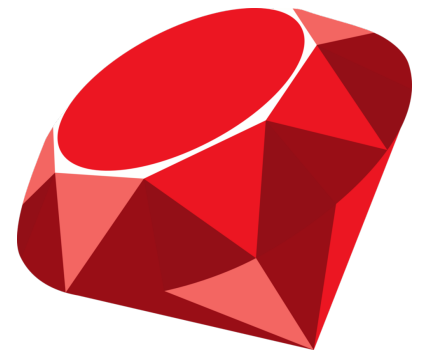
4. User management

5. Debugging

Introduction to Rails

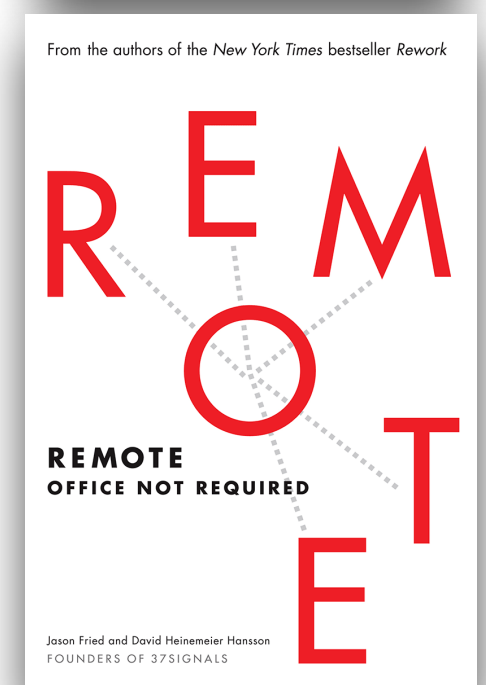
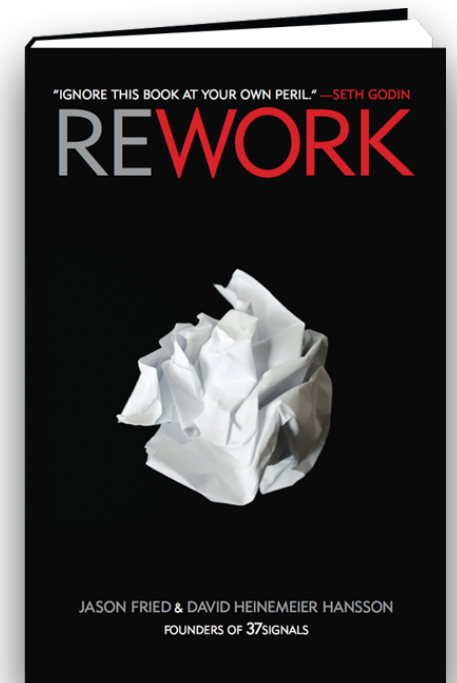


What is Rails?

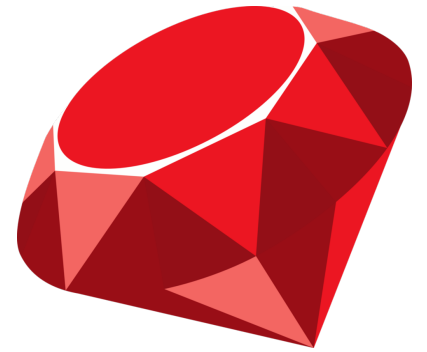


Web application
development framework
written in Ruby

Created in 2004 by
DHH (*David Heinemeier
Hansson*)



What is Rails?

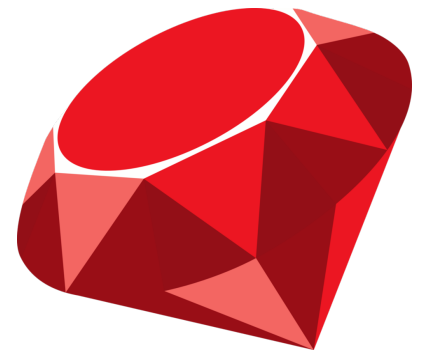


Open source: <https://github.com/rails/rails>

+ 2900 contributors

<http://rubyonrails.org>

What is Rails?

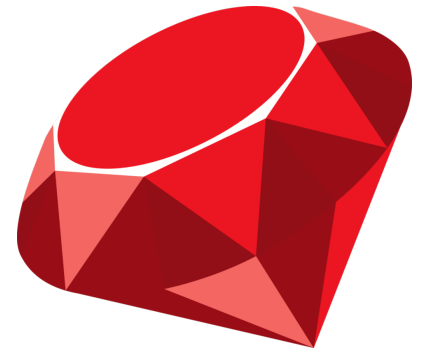


Basecamp

hulu



Why Rails?



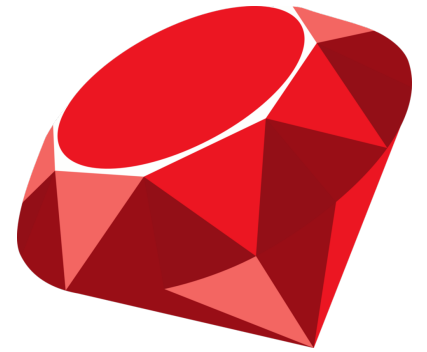
Community: A lot of nice people

A lot of gems

Nice documentation

The rails way: It just feels right

Philosophy

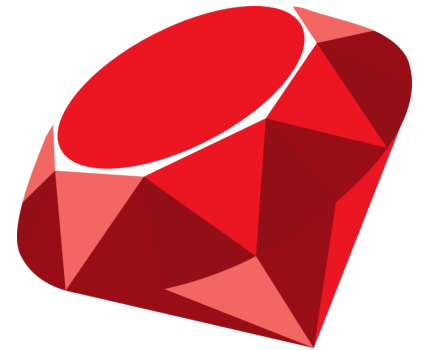


Optimize for programmer happiness

Ruby is natural

It behaves how you'd expect it to

Philosophy



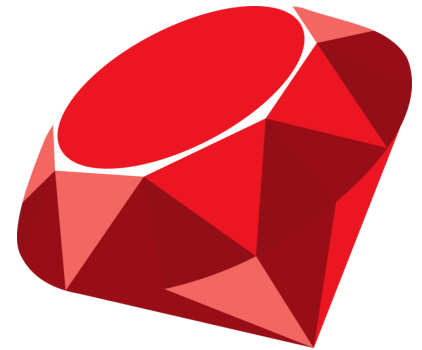
Optimize for programmer happiness

```
$ irb
$ irb(main):001:0> exit

$ irb
$ irb(main):001:0> quit
```

```
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
```

Philosophy

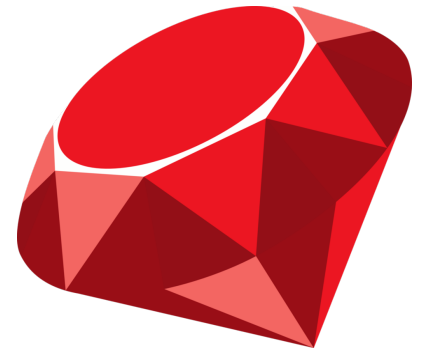


Exalts beautiful code

```
class Project < ActiveRecord::Base
  belongs_to :portfolio
  has_one :project_manager
  has_many :milestones
  has_many :deliverables, through: milestones

  validates :name, :description, presence: true
  validates :non_disclosure_agreement, acceptance: true
  validates :short_name, uniqueness: true
end
```

Philosophy



Convention over configuration

Rails has sensible defaults for just about every aspect of knitting together your application

If you need to override the conventions Rails makes that easy, too

MVC Architecture

Model-View-Controller

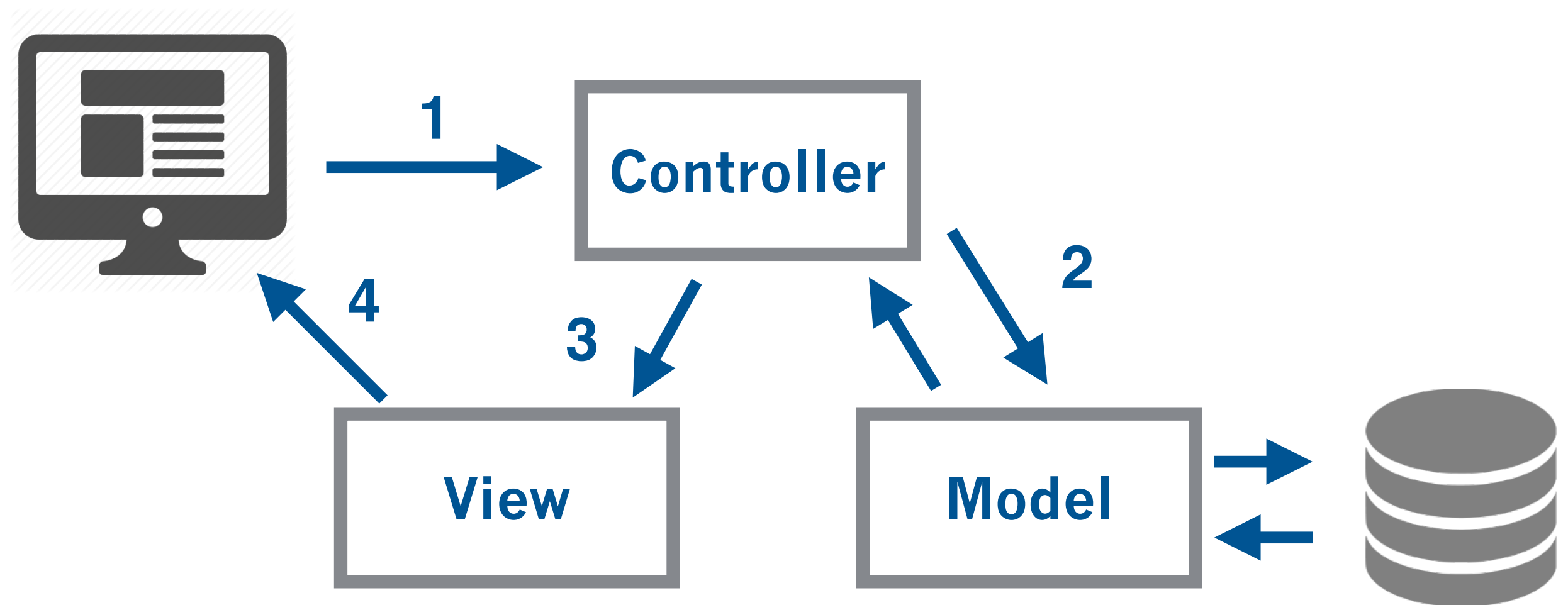
Model: It handles data and business logic

View: Responsible for generating a user interface

Controller: Orchestrate the application

MVC Architecture

Model-View-Controller



Restful Architecture

REpresentation **S**tate **T**ransfer''

The idea is to use HTTP to make calls between machines

Created by Roy Fielding in his doctoral dissertation

Platform & Language independent

Restful Architecture

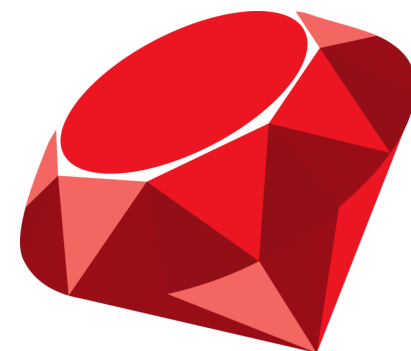
REpresentation **S**tate **T**ransfer''



Use HTTP requests for all CRUD operations

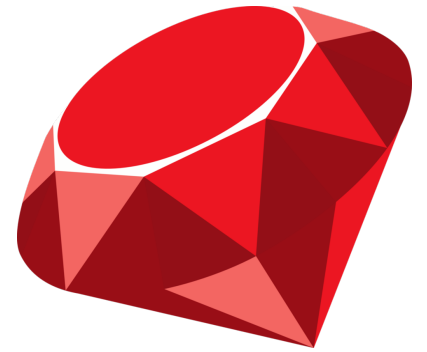
GET, POST, PUT, DELETE

```
GET /students
GET /students/1
POST /students
PUT /students/8
DELETE /students/10
```



Instant gratification

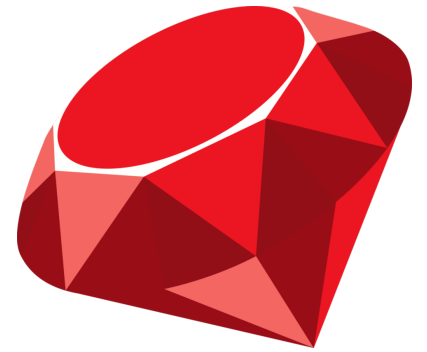
Rails setup



```
$ ruby -v  
ruby 2.2.1p85
```

```
$ rails -v  
Rails 4.2.5
```

Rails commands

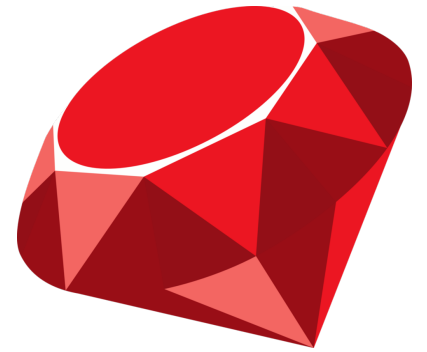


- rails new
- rails server
- rails console
- rails dbconsole
- rake
- bundle commands

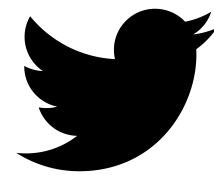
Ejercicios



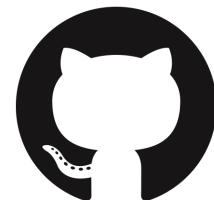
Gracias! :)



mayra@icalialabs.com



may_cabrera



mayra-cabrera

