
Medellín.rb



@RubyMedellin 
<http://medellinrb.org>

Schedule

- Say Hi
- Domain-Specific Language's in Ruby
- Yummy yummy
- Code Kata

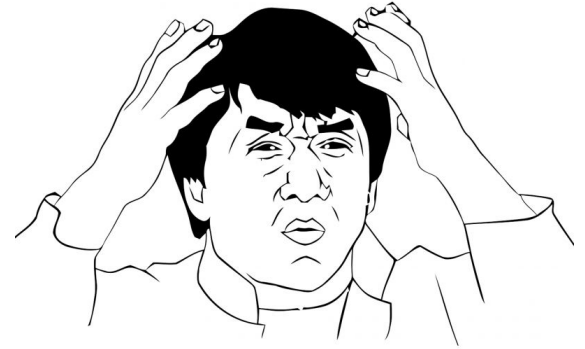
@RubyMedellin 
<http://medellinrb.org>

Domain-Specific Language

A **domain-specific language** (DSL) is a computer **language** specialized to a **particular** application **domain**. This is in contrast to a general-purpose **language** (GPL), which is broadly applicable across **domains**, and lacks specialized features for a **particular domain**.

Domain-Specific Language

A **domain-specific language** (DSL) is a computer **language** specialized to a **particular** application **domain**. This is in contrast to a general-purpose **language** (GPL), which is broadly applicable across **domains**, and lacks specialized features for a **particular domain**.



Mini-Language's

Mini-Language's


Cucumber



RSpec

```
describe 'dashboard statistics panel' do
  context 'when logged in' do
    it { is_expected.to respond_with 200 }
  end
  context 'when logged out' do
    it { is_expected.to respond_with 401 }
  end
end
```

Cucumber

Scenario: eat 5 out of 12

Given there are 12 cucumbers

When I eat 5 cucumbers

Then I should have 7 cucumbers

Scenario: eat 5 out of 20

Given there are 20 cucumbers

When I eat 5 cucumbers

Then I should have 15 cucumbers

Factory Girl

```
FactoryGirl.define do
  factory :user do
    first_name "John"
    last_name  "Doe"
    admin false
  end

  factory :admin, class: User do
    first_name "Admin"
    last_name  "User"
    admin      true
  end
end
```

ActiveRecord::Migration

```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps null: false
    end
  end
end
```

ActionDispatch::Routing

```
Rails.application.routes.draw do
  resources :products
end
```

Prefix	Verb	URI Pattern		Controller#Action
products	GET	/products	(.:format)	products#index
	POST	/products	(.:format)	products#create
new_product	GET	/products/new	(.:format)	products#new
edit_product	GET	/products/:id/edit	(.:format)	products#edit
product	GET	/products/:id	(.:format)	products#show
	PATCH	/products/:id	(.:format)	products#update
	PUT	/products/:id	(.:format)	products#update
	DELETE	/products/:id	(.:format)	products#destroy

DEMO

Yummy yummy



Code Kata



Code Kata

Kata (型 or 形 *literally: "form"*?), a Japanese word, are the detailed choreographed patterns of movements practised either solo or in pairs.

A **code kata** is an exercise in programming which helps a programmer hone their skills through practice and repetition.

Today's Kata

```
markup = FancyMarkup.new
```

```
markup.document do
  body do
    div id: 'container' do
      list class: 'pretty' do
        item "item 1"
        item "item 2"
      end
    end
  end
end
```



```
markup.to_html
```

```
"<html>
  <body>
    <div id="container">
      <ul class="pretty">
        <li>item 1</li>
        <li>item 2</li>
      </ul>
    </div>
  </body>
</html>"
```

Thanks <3



Medellín.rb

@RubyMedellin 
<http://medellinrb.org>
