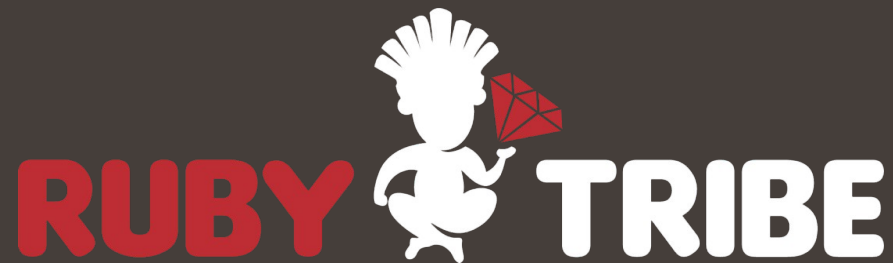


# Ruby on Rails



# Mihai Frisan

- Programator
- Indragostit de Ruby on Rails din 2006
- Sustinator al software-ului open source

Blog: [mihai.frisan.ro](http://mihai.frisan.ro)

Twitter: [@mihaifrisan](https://twitter.com/mihaifrisan)



# Yukihiro 'Matz' Matsumoto

Creator Ruby

# Ruby

- limbaj de programare orientat obiect
- dinamic
- focusat pe simplitate si productivitate
- sintaxa eleganta si naturala

# Totul in Ruby este obiect

```
# Tipareste "MAJUSCULE"  
puts "majuscule".upcase  
  
# Tipareste valoarea absoluta a lui -5  
puts -5.abs  
  
# Tipareste de 5 ori "Ruby Rocks!"  
5.times do  
  puts "Ruby Rocks!"  
End  
  
# Tipareste numerele de la 1 la 100  
1.upto(100) { |i| puts i }
```

## Ruby este un limbaj de programare dinamic

```
def len(list)
  x = 0
  list.each do |element|
    x += 1
  end
end
```

```
public static int len(List list)
{
  int x = 0;
  Iterator listIterator =
  list.iterator();
  while(listIterator.hasNext()){
    x += 1;
  }
}
```

## Ruby este focusat pe simplitate si productivitate

```
class Circle
  attr_accessor :center, :radius
end
```

```
Class Circle
  private Coordinate center, float radius;
  public void setCenter(Coordinate center){
    this.center = center;
  }
  public Coordinate getCenter(){
    return center;
  }
  public void setRadius(float radius){
    this.radius = radius;
  }
  public Coordinate getRadius(){
    return radius;
  }
}
```

# Ruby on Rails





# David Heinemeier Hansson

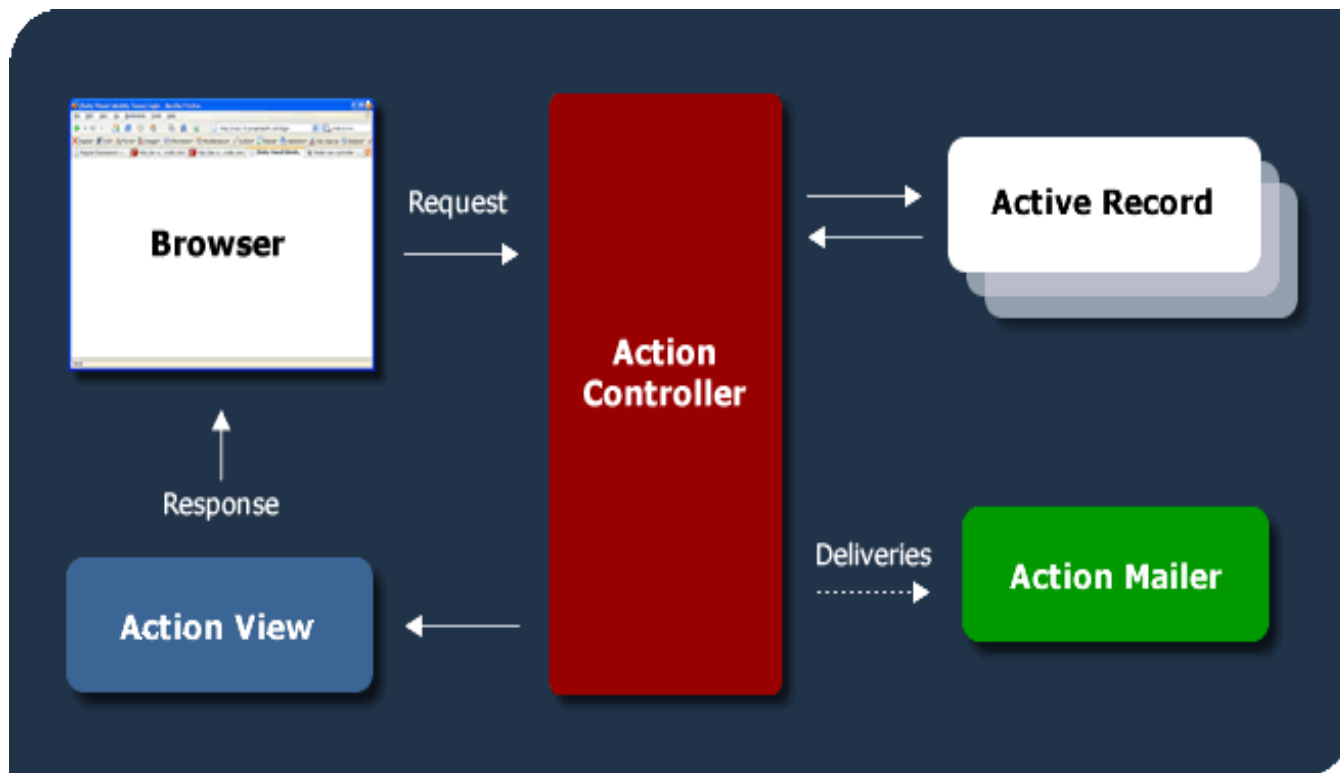
Creator Ruby on Rails

# Rails



- framework destinat dezvoltarii de aplicatii Web
- full-stack framework
- MVC pattern
- Convention over Configuration

# Arhitectura Rails



# Models



# ActiveRecord

- Object Relation Mapping (ORM)
- Mapeaza tabele la clase si randurile la obiecte
- Database Agnostic

# ActiveRecord

```
CREATE TABLE entries (  
  id int(11) NOT NULL auto_increment,  
  name varchar(255),  
  comments varchar(255),  
  PRIMARY KEY (id)  
);
```

```
class Entry < ActiveRecord::Base  
  # nothing here!  
end  
  
entry = Entry.new  
entry.name = "John"  
entry.comments = "What a great Web site!"  
entry.save  
  
entry = Entry.find(1)  
puts entry.name # output: "John"
```

# ActiveRecord

```
class Entry < ActiveRecord::Base
  belongs_to :account
  has_many :comments

  validates_presence_of :name

  acts_as_taggable
end

entry = Entry.new
entry.name = "John"
entry.comments << Comment.new("What a great Web site!")
entry.save

Entry = Entry.find_by_name("John")
```

## Rails programmers like fat Models





# Controler



# ActionController

- Managementul request-urilor
- Mapeza actiunile la metode
- Separa logica de prezentare
- Implementeaza resursele RESTful

# ActionController

```
class ProductsController < ApplicationController
  # GET /products
  # GET /products.xml
  def index
    @products = Product.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @products }
    end
  end

  # GET /products/1
  # GET /products/1.xml
  def show
    @product = Product.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @product }
    end
  end

  # GET /products/new
  # GET /products/new.xml
  def new
    @product = Product.new

    respond_to do |format|
```

# View



# ActionView

- Prezinta utilizatorului datele
- Foloseste template-uri
- Mecanism de caching

# ActionView

```
<p>  
  <b>Name:</b>  
  <%=h @product.name %>  
</p>  
  
<%= link_to 'Edit', edit_product_path(@product) %> |  
<%= link_to 'Back', products_path %>
```

# Proiecte dezvoltate in Rails



43Things



***shopify***

**unfuddle™**



**Basecamp**



**mindmeister**

# Companii care folosesc Rails

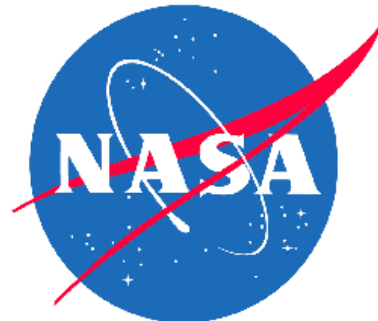
amazon.com.



YAHOO!



 Capgemini  
CONSULTING. TECHNOLOGY. OUTSOURCING



ORACLE®

 EA  
ELECTRONIC ARTS™

SIEMENS



# De ce as folosi Rails?

- Convention over Configuration
- Data mapping
- AJAX
- Testing Framework
- Agile Development
- Numar impresionant de librarii