

# Ruby on Rails

Lugano, 29.03.2014



[railsgirlsticino@gmail.com](mailto:railsgirlsticino@gmail.com) | [@RailsGirls\\_TI](https://twitter.com/RailsGirls_TI) | [WWW.RAILSGIRLS.COM/TICINO](http://WWW.RAILSGIRLS.COM/TICINO)



- ◆ INTRODUCTION
- ◆ THE BASICS
- ◆ THE BASICS – ROR
- ◆ DEVELOPING THE APPLICATION
- ◆ CONCLUSIONS




# 01. Introduction

# Introduction – The Background



◆ The programming language Ruby was created by Yukihiro Matsumoto in 1993



キヤ kya	キュ kyu	キョ kyo
シャ sha	シュ shu	ショ sho
チャ cha	チュ chu	チョ cho
ニヤ nya	ニュ nyu	ニョ nyo
ヒヤ hya	ヒュ hyu	ヒョ hyo
ミヤ mya	ミュ myu	ミョ myo
リヤ rya	リュ ryu	リョ ryo
ギヤ gya	ギュ gyu	ギョ gyo
ジャ ja	ジュ ju	ジョ jo
ビヤ bya	ビュ byu	ビョ byo
ピヤ pya	ピュ pyu	ピョ pyo

◆ But initially it wasn't a great success!

- Unpopular for over a decade
- Originally the documentation only existed in Japanese!!!

◆ The turning point was in 2006

- The year when the framework Ruby on Rails started gaining recognition



---

# Introduction – Some Features



The programming language Ruby is:

- ◆ open source
- ◆ general purpose
- ◆ interpreted
- ◆ object oriented

Ruby has borrowed a lot from other programming languages, such as:  
Smalltalk, Perl, Python, C, C++, PHP, Phyton...

<http://www.ruby-lang.org/en/documentation/ruby-from-other-languages/>

In fact, one of their first slogans was:

***( Smalltalk + Perl ) / 2***

# Introduction – What is Ruby On Rails (ROR) ?



## RUBY ON RAILS IS

“... an Open-source web framework that’s optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration.”

“... a breakthrough in lowering the barriers of entry to programming. Powerful web applications that formerly might have taken weeks or months to develop can be produced in a matter of days.”

( source: <http://www.rubyonrails.org> )

- 💎 Ruby on Rails, or RoR , is an open source MVC (Model View Controller) framework based on Ruby.
- 💎 With Ruby on Rails , you can develop applications in a simple way minimising the part of coding that is usually repeated in every application.
- 💎 The pattern Model View Controller is a programming convention that makes the distinction between the data representation, the logic of the application and the content more clear.
- 💎 The applications developed in Ruby on Rails are especially suited for dynamic, flexible use with a continuous need for updating or future additions.

---

# Introduction - Why Ruby On Rails

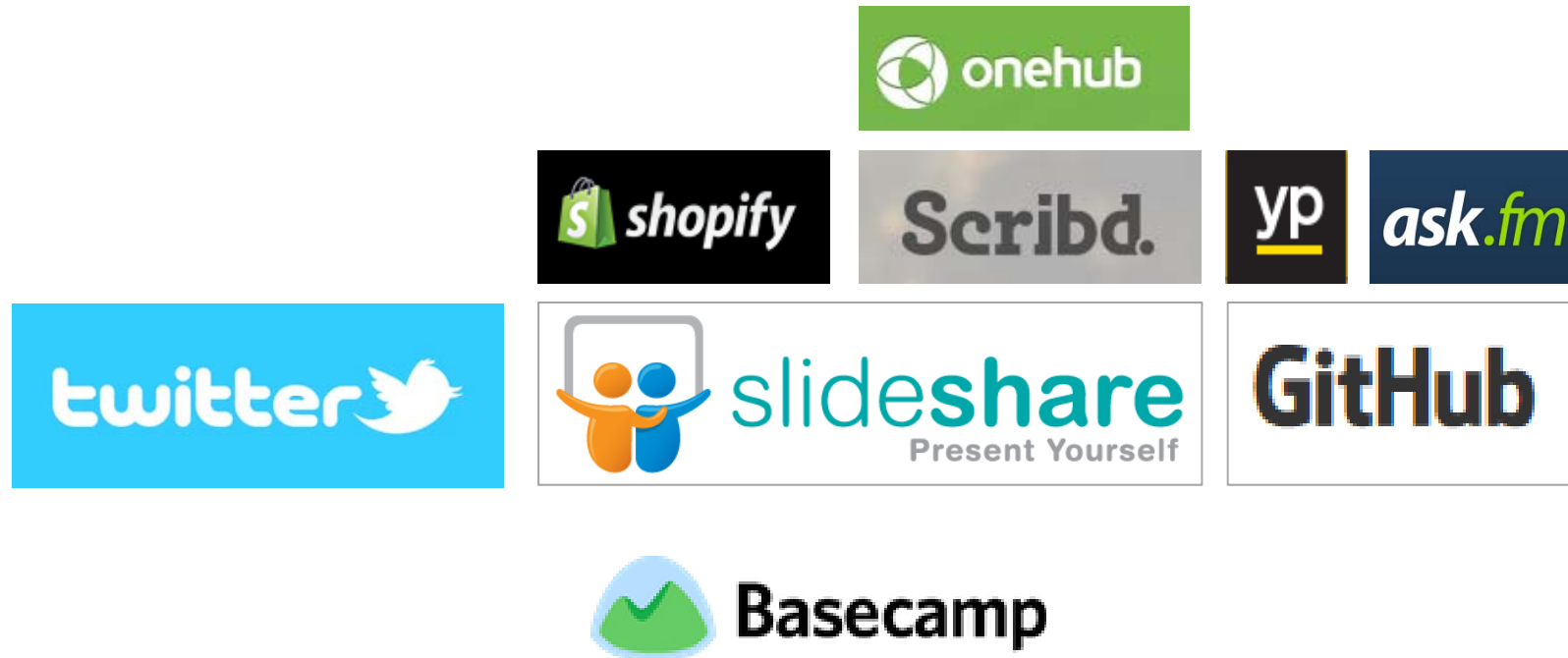
*The strengths of RoR*



## WHY CHOOSE RUBY ON RAILS FOR DEVELOPING?

- 💎 Simple
- 💎 Intuitive
- 💎 Scalable
- 💎 Fast
- 💎 Minimises code complexity
- 💎 The code is self explanatory
- 💎 Since it eliminates the dull parts of coding, there's more time to concentrate on the essentials: the user's idea of how the app should be.

# Introduction – Websites On Rails



Twitter, SlideShare, GitHub, Basecamp, Shopify, Scribd, OneHub, Yellow Page, Ask.fm, Cookpad, Hulu, CrunchBase, Zendesk, MedHelp, Freckle, CrazyEgg, 43Things.....



# Introduction – Who uses Rails





## 02. The basics

---

# Object Oriented Programming (OOP)

*The definition*



Allows grouping definitions of data structures and procedures operating on them into restricted bits (classes) of the source code.

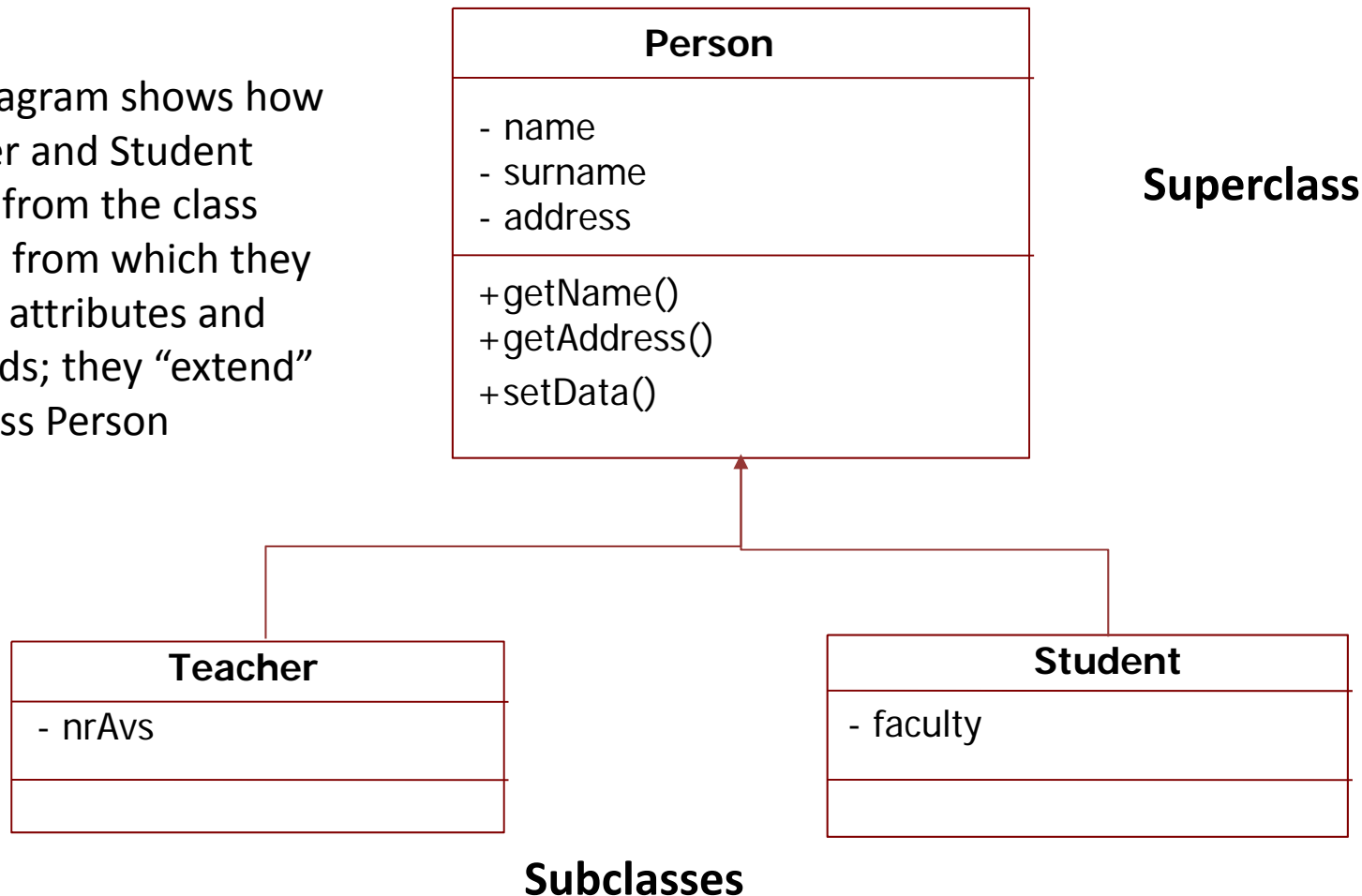
- ◆ In object oriented programming (OOP, Object Oriented Programming) an object may be seen as a container of data with attributes (data) and a group of methods (procedures) defined in its class.
- ◆ An object is an instance of a class
  - It is assigned all the attributes and methods defined in its class.
- ◆ Features :
  - ◆ Heritability
  - ◆ Polymorphism
  - ◆ Encapsulating

# Object Oriented Programming (OOP)

## Heritability



This diagram shows how Teacher and Student derive from the class Person from which they inherit attributes and methods; they “extend” the class Person



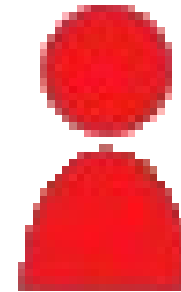
# Object Oriented Programming (OOP)

*Example*



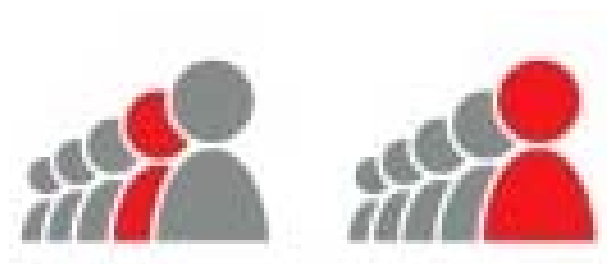
```
class Person
  attr_accessor :name, :surname, :address

  def initialize (name, surname, address)
    @name = name
    @surname = surname
    @address = address
  end
end
```



*Object*

*Class*





## 03. The basics - ROR

# MVC

*The Model View Controller pattern*



The data (model) is separated from the interface (view)

## Model

- Maintains the connection between the objects and the database, and is in charge of confirming, connecting, transactions and other processes.
- Access to the data and the logic of the application
- Independent of the view and the controller.

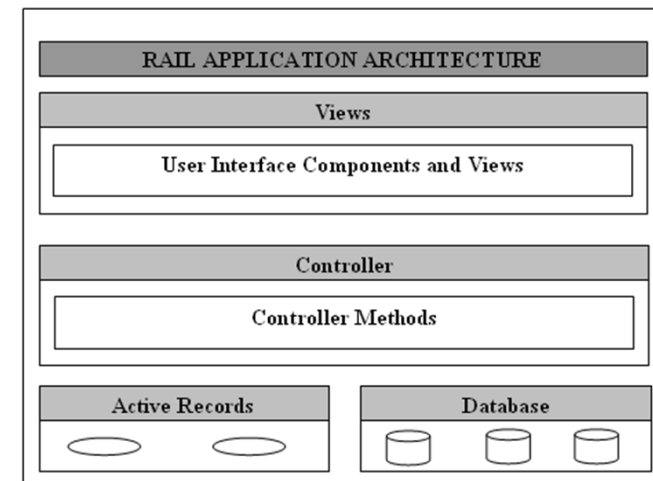
## View

- Presentation of the data and interaction with the user
- Only displays what is in the model

## Controller

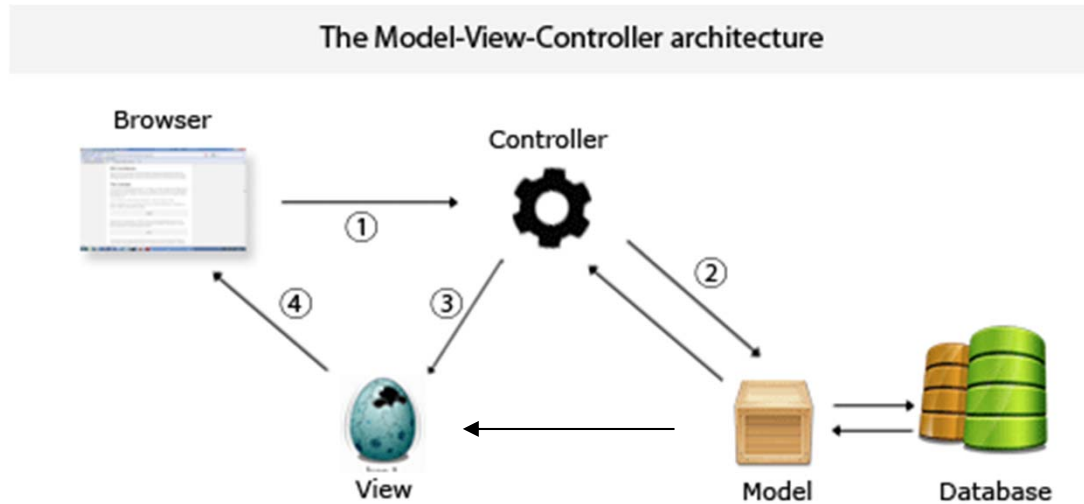
The logic of the application.

- Coordinates the interaction between the user, the view and the model.
- In charge of what is happening



# MVC

## How It Works



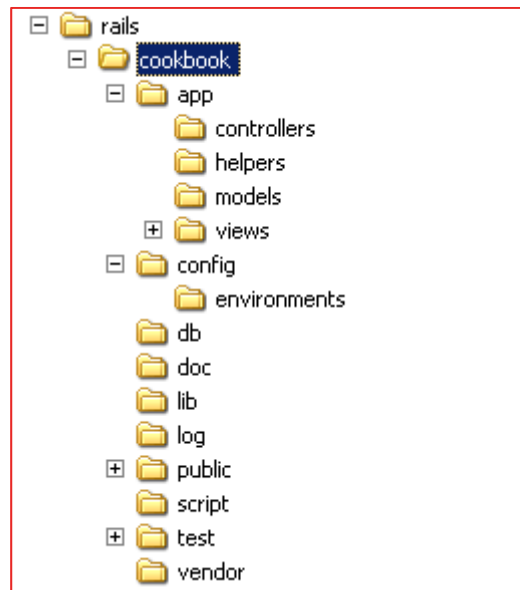
1. The browser sends a request
2. The controller interacts with the model
3. The controller calls the view
4. The view displays the information in the browser



# The Structure Of A Ruby on Rails Application



- ❖ The applications developed using Rails share a common feature; they are all organised according to the same structure.
- ❖ This is because the command rails generates a group of directories and files that follow a guideline that permits Rails to implement various things automatically (for instance loading and generating files as well as recognising them during runtime).
- ❖ This common structure also aids in understanding other developers' code easily, as the projects are organised according to the same principles.
- ❖ The structure of the applications



# Naming conventions – Ruby on Rails



Rails has configuration conventions to save the developer the trouble of having to specify e.g. the association between tables and classes.

For the association between tables and classes to work, it's enough that the tables are given the pluralised lowercase names of the classes.

EXAMPLE : The table messages thus mapped to the class Message, authors to Author and topics to Topic. The mechanism of the ActiveRecord that takes care of the conversion is more intelligent than you'd think; for instance, it understands that the plural of "person" is "people" and the plural of "status" is "statuses".

## Models

<b>Tabella</b>	products
<b>File</b>	app/models/product.rb
<b>Classe</b>	Product

## The Controller

<b>URL</b>	http://../store/list
<b>File</b>	app/controllers/store_controller.rb
<b>Classe</b>	StoreController
<b>Metodo</b>	list
<b>Layout</b>	app/views/layouts/store.html.erb

## Views

<b>URL</b>	http://../store/list
<b>File</b>	app/views/store/list.html.erb (o .builder o .rjs)
<b>Helper</b>	modulo StoreHelper
<b>File</b>	app/helpers/layouts/store_helper.rb



# 04. Developing the application

<http://guides.railsgirls.com/app/>



## 05. Conclusions

---

# Tutorial



<http://railsforzombies.org/>

<http://ruby.railstutorial.org/>

<http://railscasts.com/>

<http://guides.rubyonrails.org/>

---

# Conclusions



# QUESTIONS

---

# THANK YOU



<http://railsgirls.com/ticino>

[email] [railsgirlsticino@gmail.com](mailto:railsgirlsticino@gmail.com)

[twitter] [@RailsGirls\\_TI](https://twitter.com/RailsGirls_TI)

[facebook] [Rails Girls Ticino](https://www.facebook.com/RailsGirlsTicino)

---