Happy Coding with

Ruby on Rails

Maahai L.Ochirkhuyag

2009-6-6 @MAJ Meeting, Tokyo

Ochirkhuyag.is_who?

ochko.blogspot.com
maaxai.blogspot.com

2000~ programming
2005~ USI Inc.
2008~2010 Waseda Univ. Master's course

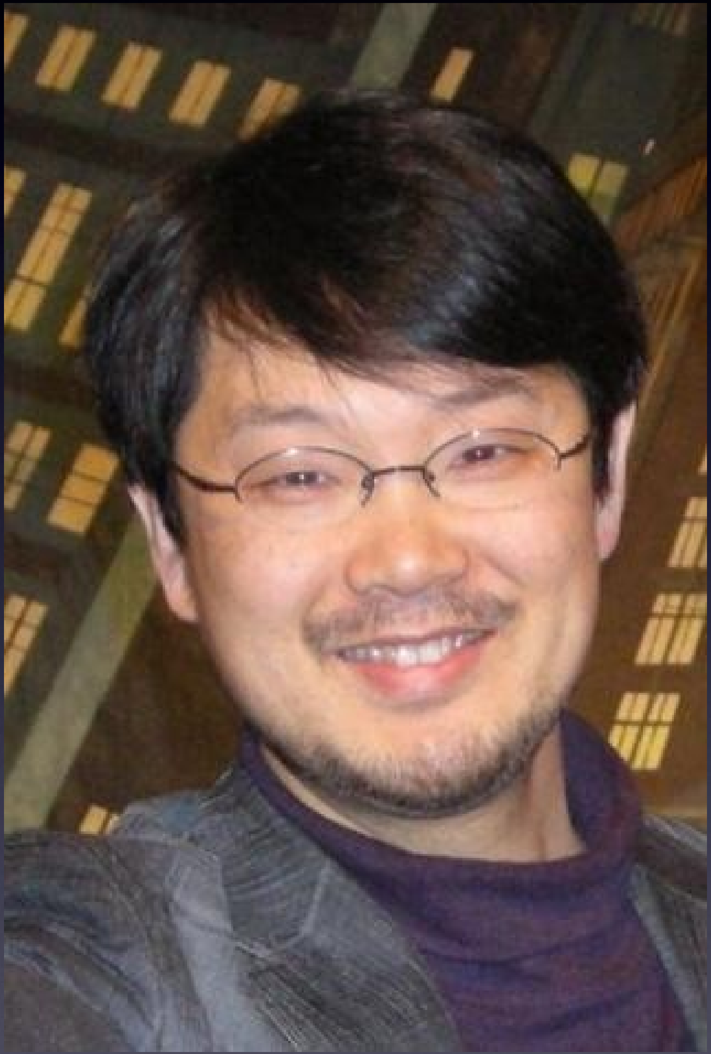
2007~ Ruby & RoR

Ruby

“more powerful than Perl, and more OO than Python”

“interpreted, object-oriented, portable, untyped, garbage collected”

Yukihiro ‘Matz’ Matsumoto, creator of Ruby:



"Often people, especially computer engineers, focus on the **machines**. They think, "By doing this, the machine will run faster. By doing this, the machine will run more effectively. By doing this, ... But in fact we need to focus on **humans**, on how humans care about doing programming or operating the application of the machines. We are the masters. They are the slaves.”

Everything is Object

```
>> 2009.class              # → Fixnum

>> :lets_start.class       # → Symbol

>> "ruby is sucking".capitalize
# → "Ruby is sucking"

>> 3.times {puts "Сайн уу?"}
Сайн уу?
Сайн уу?
Сайн уу?
```

Ruby array is *Powerful*

```
>> a = [ 1, 2, 3, 'four', 'v', nil, 2 ]
=> [1, 2, 3, "four", "v", nil, 2]
>> a.compact.uniq
=> [1, 2, 3, "four", "v"]
>> [1, 2, 3, 4, 5, 6].select { |e| e % 2 == 0 }
=> [2, 4, 6]
>> [1, 2, 3, 4].inject(0) { |res, el| res + el }
=> 10
```

Ruby classes are *Open*

```
# Numeric is ruby's built-in class
# We can add a method to it

class Numeric
  def plus(x)
    self.+(x)
  end
end

5.plus 6      # → 11
```

Block is *Expressive*

```
search_engines =
  %w[Google Yahoo MSN].map do |engine|
    "http://www." + engine.downcase + ".com"
  end

# → [ "http://www.google.com",
      "http://www.yahoo.com",
      "http://www.msn.com" ]
```

Built-in Regular expression

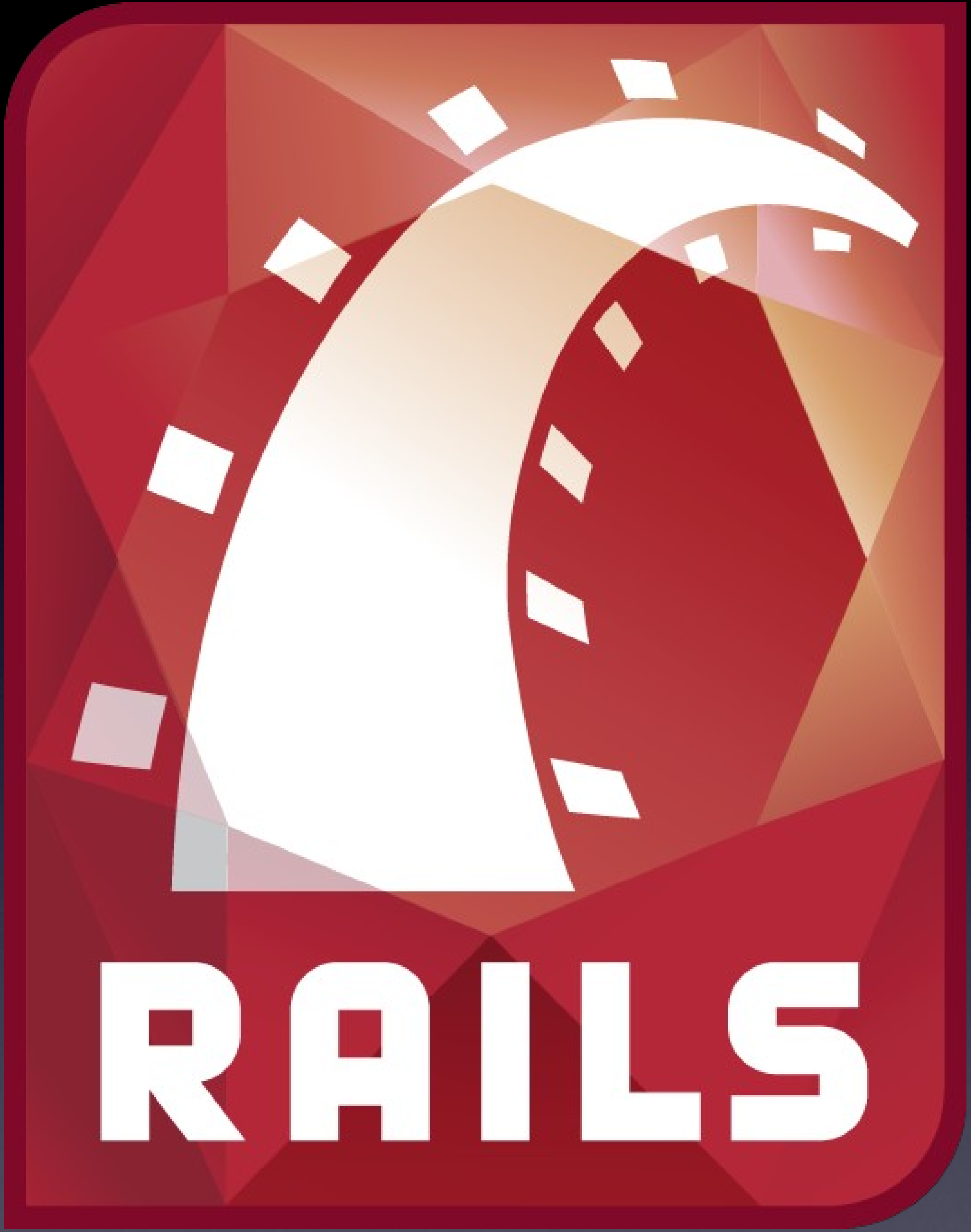
```
//.class
# → Regexp

a = "the quick brown fox"

a =~ /q[uia]*ck/
# → 4

a.gsub(/[aeiou]/, '*')
# → "th* q**ck br*wn f*x"
```

Mixin

```
class MyArray
  include Enumerable
  # implement each
end

m = MyArray.new()
m << 1 << 2 << 3
m.find{|e| e % 2 == 0}
# → 2
```

try ruby! (in your browser)

http://tryruby.hobix.com/


RAILS

David ‘dhh’ Heinemeier Hansson, creator of Ruby on Rails

- Convention over Configuration
- DRY
- High Productivity

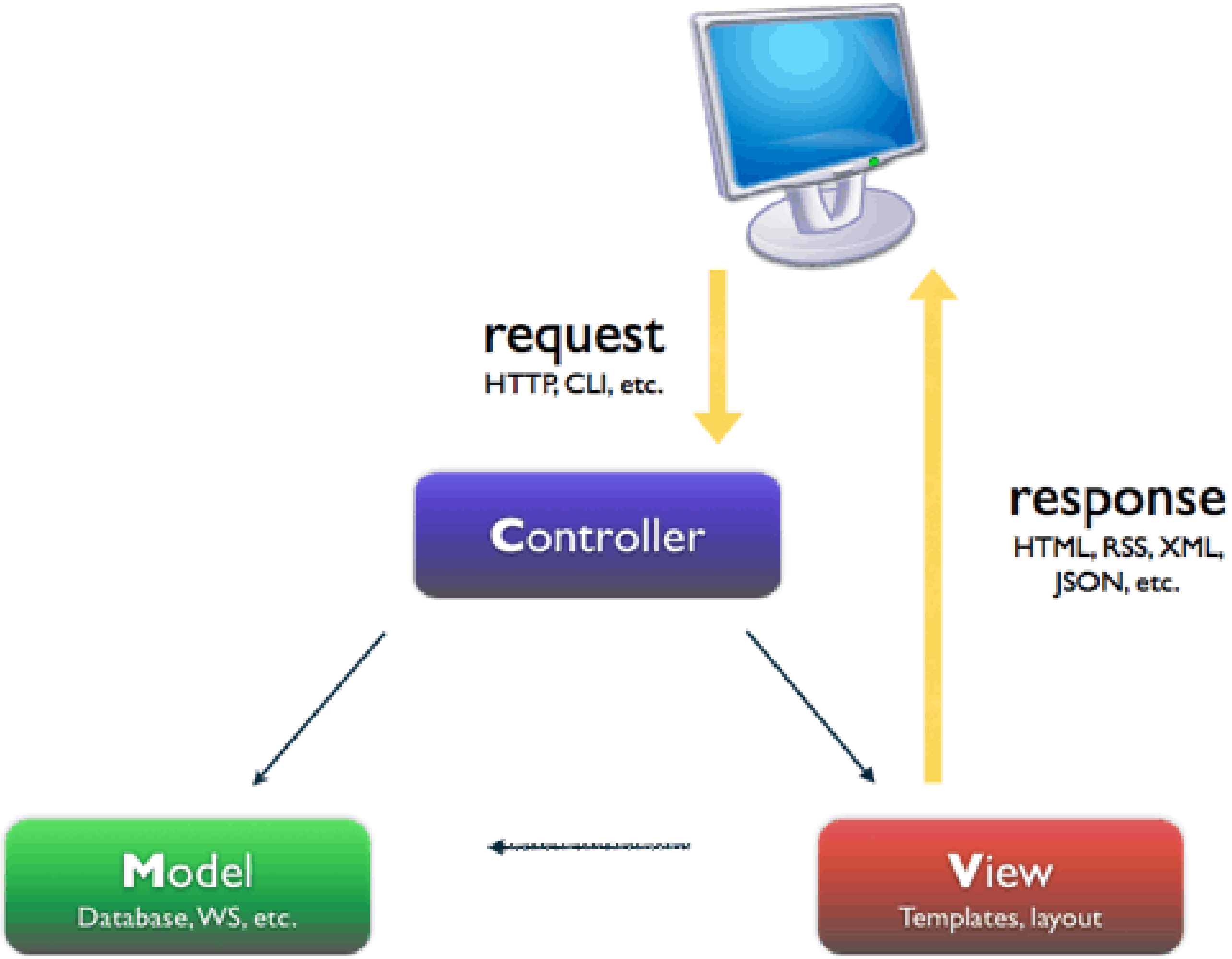
Websites on Rails

Basecamp™
Project Collaboration
twitter
Backpack®
YELLOW PAGES™
github
SOCIAL CODE HOSTING
shopify
e-commerce made easy
ODEO
Geni
everyone's related
Lighthouse
beautifully simple issue tracking

Ruby on Rails vs...

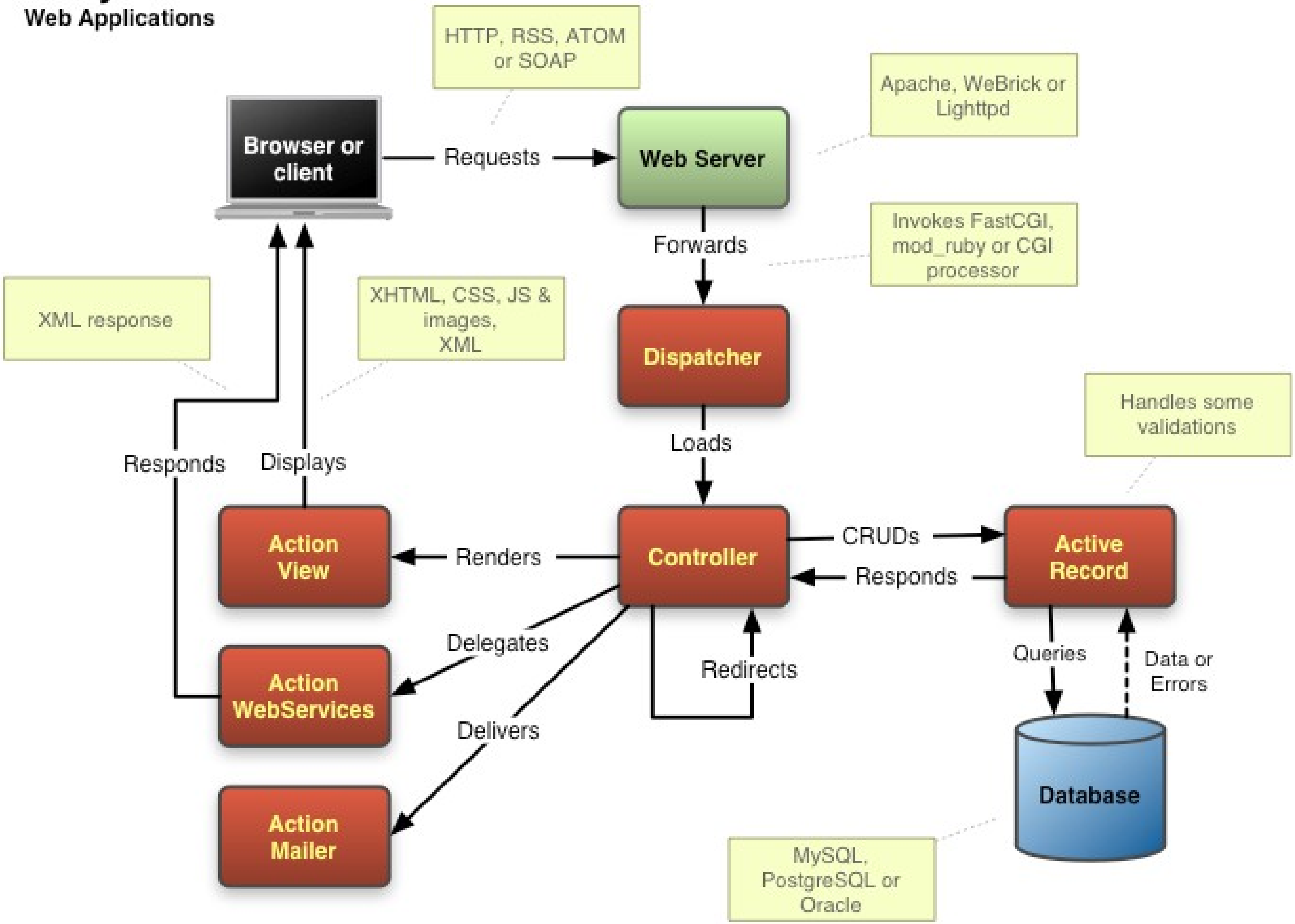
Java	Python	PHP	Ruby
Struts	django	CodeIgniter®	RAILS
Spring	Pylons	CakePHP	ramaze
tapestry	ZOPE Community		Sinatra
etc.	etc.	etc.	etc.

MVC Architecture



Ruby on Rails

Web Applications



HTTP, RSS, ATOM or SOAP

Apache, WeBrick or Lighttpd

Browser or client

Requests

Web Server

Forwards

Invokes FastCGI, mod_ruby or CGI processor

XML response

XHTML, CSS, JS & images, XML

Dispatcher

Loads

Handles some validations

Responds

Displays

Action View

Renders

Controller

CRUDs

Responds

Active Record

Delegates

Action WebServices

Redirects

Queries

Data or Errors

Delivers

Database

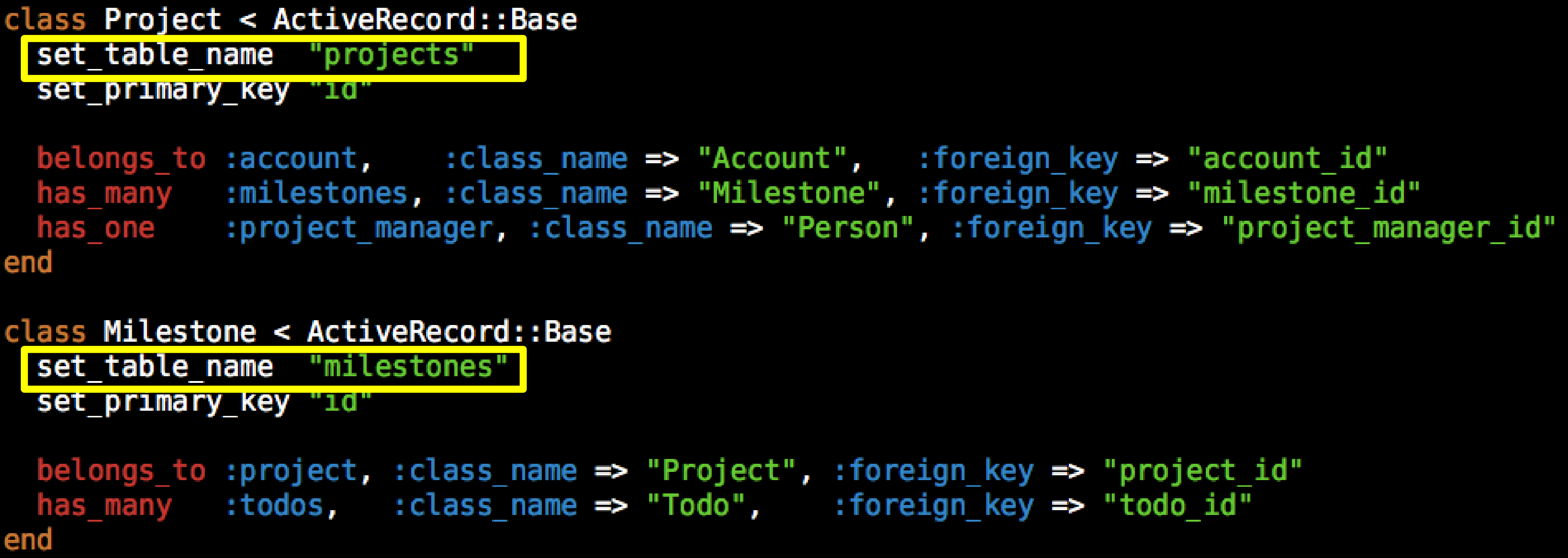
Action Mailer

MySQL, PostgreSQL or Oracle

Convention over Configuration

CoC

```
class Project < ActiveRecord::Base
  set_table_name  "projects"
  set_primary_key "id"

  belongs_to :account,    :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  set_table_name  "milestones"
  set_primary_key "id"

  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```

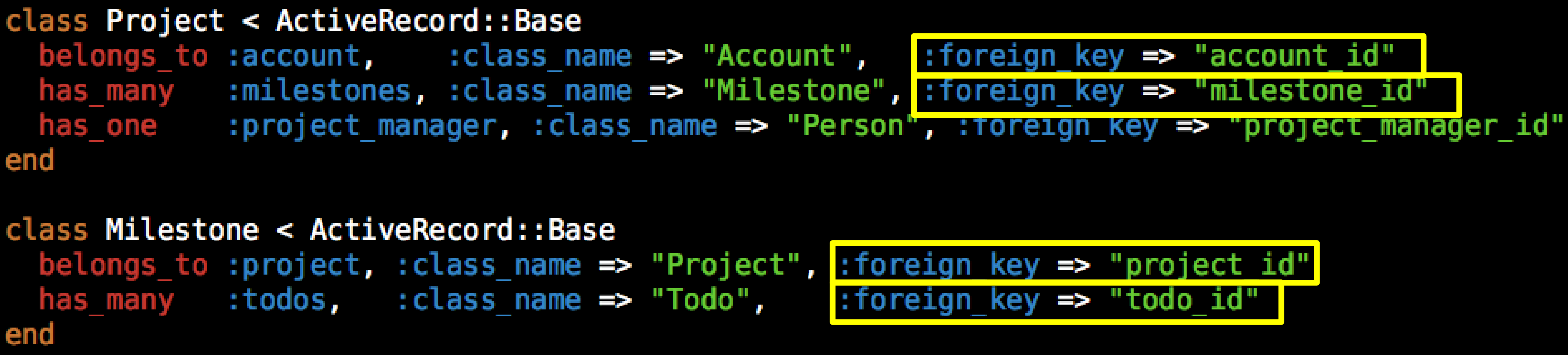


```
class Project < ActiveRecord::Base
  set_table_name  "projects"
  set_primary_key "id"

  belongs_to :account,   :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  set_table_name  "milestones"
  set_primary_key "id"

  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```

```
class Project < ActiveRecord::Base
  set_primary_key "id"

  belongs_to :account,    :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  set_primary_key "id"

  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```

```ruby
class Project < ActiveRecord::Base
  set_primary_key "id"

  belongs_to :account,    :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  set_primary_key "id"

  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```

```ruby
class Project < ActiveRecord::Base
  belongs_to :account,    :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```



```
class Project < ActiveRecord::Base
  belongs_to :account,    :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```

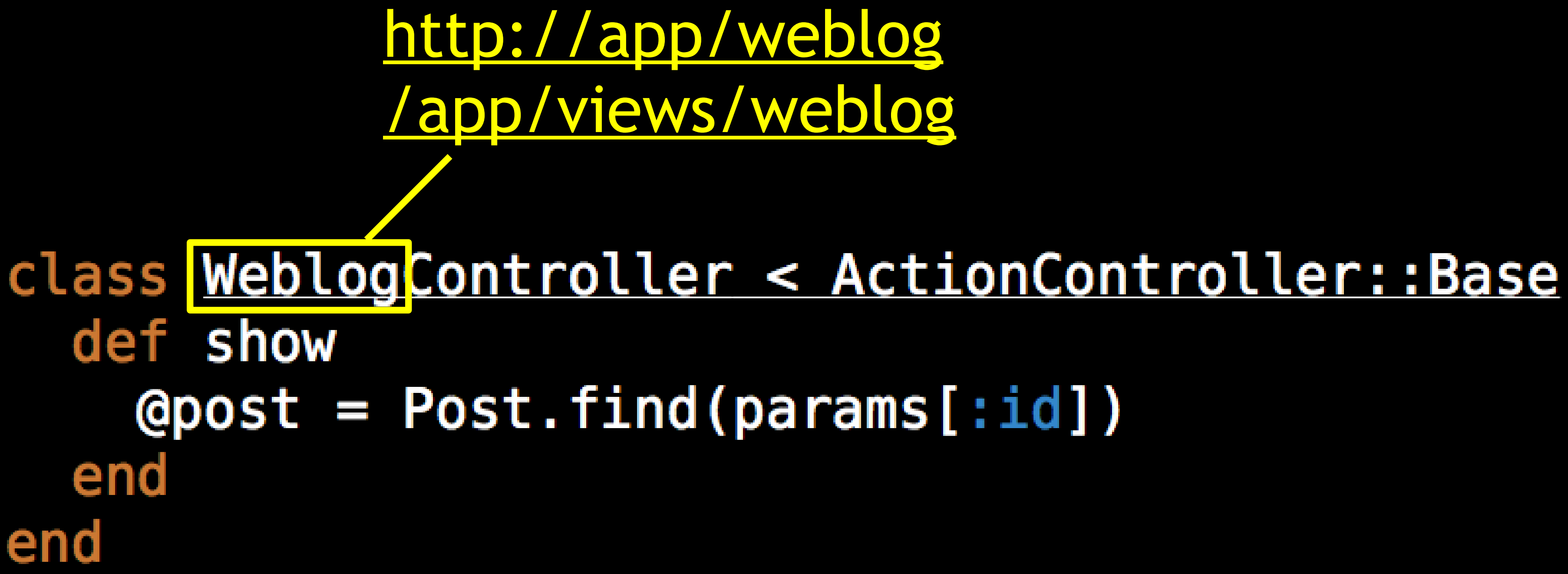
```
class Project < ActiveRecord::Base
  belongs_to :account,    :class_name => "Account"
  has_many   :milestones, :class_name => "Milestone"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  belongs_to :project, :class_name => "Project"
  has_many   :todos,   :class_name => "Todo"
end
```

```ruby
class Project < ActiveRecord::Base
  belongs_to :account,    :class_name => "Account"
  has_many   :milestones, :class_name => "Milestone"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  belongs_to :project, :class_name => "Project"
  has_many   :todos,   :class_name => "Todo"
end
```

```
class Project < ActiveRecord::Base
  belongs_to :account
  has_many   :milestones
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  belongs_to :project
  has_many   :todos
end
```

```ruby
class Project < ActiveRecord::Base
  set_table_name  "projects"
  set_primary_key "id"

  belongs_to :account,    :class_name => "Account",   :foreign_key => "account_id"
  has_many   :milestones, :class_name => "Milestone", :foreign_key => "milestone_id"
  has_one    :project_manager, :class_name => "Person", :foreign_key => "project_manager_id"
end

class Milestone < ActiveRecord::Base
  set_table_name  "milestones"
  set_primary_key "id"

  belongs_to :project, :class_name => "Project", :foreign_key => "project_id"
  has_many   :todos,   :class_name => "Todo",    :foreign_key => "todo_id"
end
```

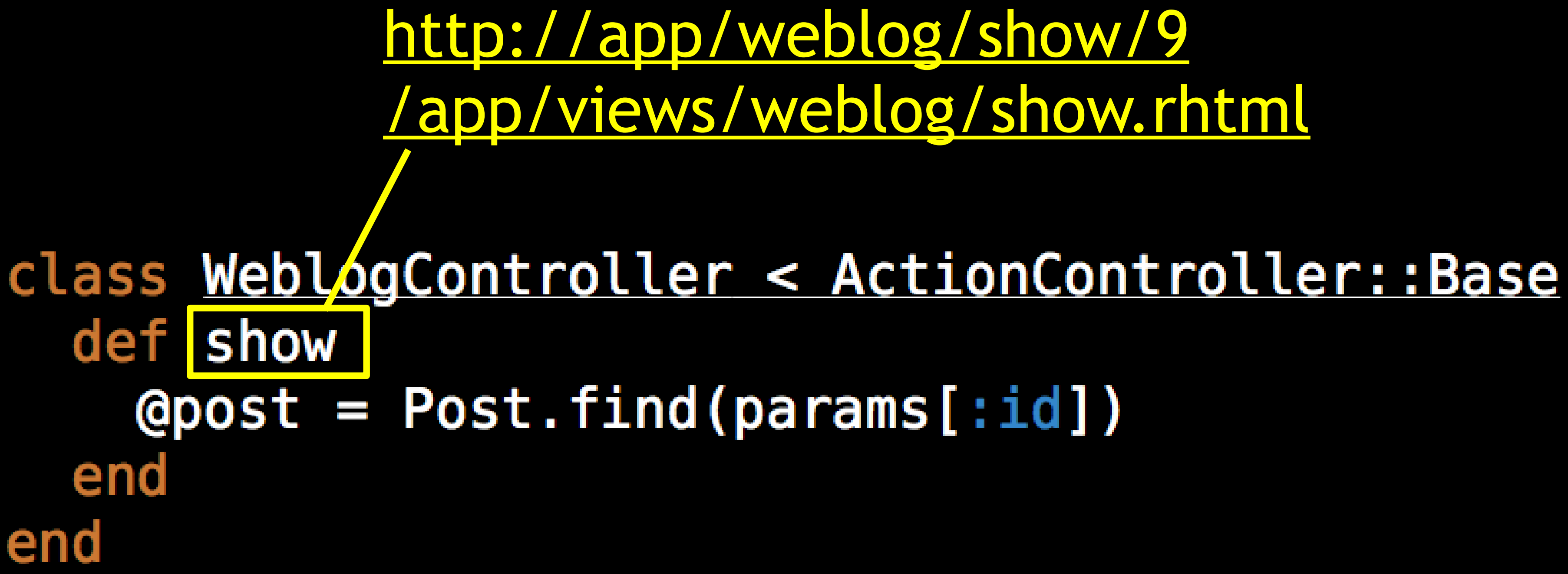
```ruby
class WeblogController < ActionController::Base
  def show
    @post = Post.find(params[:id])
  end
end
```



http://app/weblog

/app/views/weblog

```
class WeblogController < ActionController::Base
  def show
    @post = Post.find(params[:id])
  end
end
```



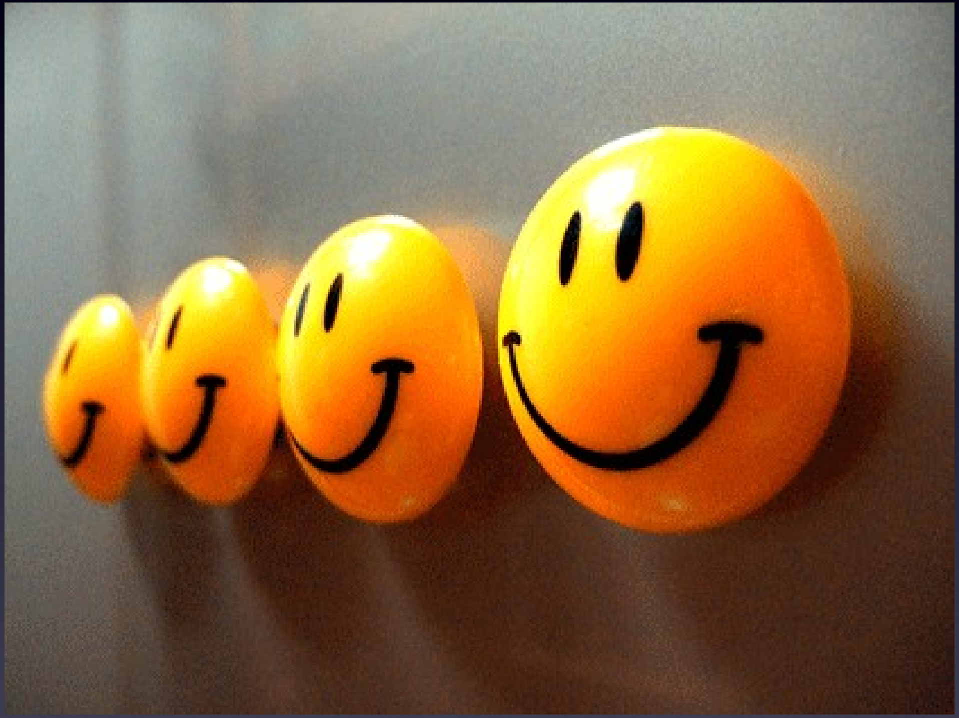
http://app/weblog/show/9

/app/views/weblog/show.rhtml

```
class WeblogController < ActionController::Base
  def show
    @post = Post.find(params[:id])
  end
end
```

Don't Repeat Yourself(DRY)

Productivity ← Motivation ← Happiness ← Beautiful Code



Happy Coding Demo

Q&A

?

```
if you.want? contact_me_by  {
:twitter => ochko,
:github => ochko }
```

References

http://marklunds.com/rails101

http://www.slideshare.net/Joopp/ruby-on-rails-presentation-presentation

http://media.rubyonrails.org/presentations/programminghappiness.pdf

http://media.rubyonrails.org/video/rails_blog_2.mov