

# Starting up with **Ruby on Rails**

Stories from a startup

Cyril David

Miguel Vega

Philippine Open Source Conference 2006

# PHP



“Start at the very beginning, the very best place to start”

- Julie Andrews, *Mary Poppins*

**Easy to Learn**

```
<?php echo("hello world!");?>
```

```
<?php include('header.php'); ?>.
```

```
.
```

```
<?php echo('Welcome to My Personal Homepage!'); ?>.
```

```
.
```

```
<?php include('footer.php'); ?>.
```

**Discipline?**

```
<?php.  
// displaying all users.  
echo "  
<table>.  
  <tr>.  
    <td>username</td>.  
    <td>fullname</td>.  
  </tr>";  
  
.  
$sql = "select * from users";  
$result = mysql_query($sql);  
while($user = mysql_fetch_array($result)) {  
  echo "  
    <tr>.  
      <td>{$user['username']}</td>.  
      <td>{$user['fullname']}</td>.  
    </tr>";  
}  
  
.  
echo "  
</table>";  
?>
```

```
<?php .
.
class User.
{.
    function getUser() .
    {
        $sql = "SELECT * from Users WHERE username = '{$_POST['username']}' " .
        /* ...DB CODE HERE... */.
    } .
    .
    function currentUser().
    {
        $sql = "SELECT * from Users WHERE id = '{$_SESSION['user_id']}' " .
        /* ...DB CODE HERE... */.
    } .
} .
?>.
```



**Structure?**

- images
- AddCompletedProject.php
- AddCompletedProject.tpl
- AddOngoingProject.php
- AddOngoingProject.tpl
- AddProposal.php
- AddProposal.tpl
- EditCompletedProject.php
- EditCompletedProject.tpl
- EditOngoingProject.php
- EditOngoingProject.tpl
- EditProject.php
- EditProject.tpl
- EditProposal.php
- EditProposal.tpl
- EditProposal.tpl-org
- EncodeProposal.php
- EncodeProposal.tpl
- ManagementModule.php
- MyDrafts.php
- MyDrafts.tpl
- MyOngoingProjects.php
- MyOngoingProjects.tpl
- MyProposals.php
- MyProposals.tpl
- ProgressUpdate.php
- ProgressUpdate.tpl
- Project.php
- ProjectAccounting.php
- ProjectAccounting.tpl
- ProjectInformation.php

**Framework?**

- **Prado**
- **CakePHP**
- **Symfony**
- **Zend Framework**
- **etc...**

```
public function login($username,$password=null).
{.
    $username = addslashes($username);.
    $password = addslashes($password);.

    if(is_null($password)).
        $authenticated=false;.
    else.
    {.

        //$db = new TAdodb;
        $sql = "SELECT * FROM users WHERE username='$username' AND password='$password'";.
        $result=$this->db->Execute($sql);.
        $authenticated=$result->RecordCount()>0;.
    }.
    return $authenticated;.
} .
```

```

public function onInit($param).
{
    $this->dataBind();
    $this->userinfo = $this->Module->getUserInfo($_GET['id']);
    $this->profile = $this->Module->getMyProfile($this->userinfo['faculty_id']);
    $this->history = $this->Module->getFacultyHistory($this->userinfo['faculty_id']);
    .
    .
    parent::onInit($param);
    if (!$this->IsPostBack) {
        .
        $this->setViewState('Degrees',$this->Module->getDegreeNames());
        $this->setViewState('Departments',$this->Module->getDepartmentNames());
        $this->setViewState('Ranks',$this->Module->getRankNames());
        $this->setViewState('Specialties',$this->Module->getSpecialtyNames());
        .
        $this->content->UserID->setText($this->userinfo['user_id']);
        $this->content->Username->setText($this->userinfo['username']);
        $this->content->FacultyTitle->setText($this->getProfile('fac_title'));
        $this->content->FirstName->setText($this->getProfile('first_name'));
        $this->content->MiddleInitial->setText($this->getProfile('middle_name'));
        $this->content->LastName->setText($this->getProfile('last_name'));
        $this->content->Birthdate->setText($this->getProfile('birthdate'));
        $this->content->Sex->setText($this->getProfile('sex'));
        $this->content->Category->setText($this->getProfile('category'));
        $this->content->CurrentRank->setText($this->getProfile('cur_rank'));
    }
}

```

```
<%@Page Master="MasterLayout" %>.
<com:TContent ID="content">.
<div>.
  <div class='header'>.
    My Profile.
  </div>.
  <div><%= print_r($this->User)%>.
    <table>.
      <tr>.
        <td>User ID</td>.
        <td><com:TLabel ID="UserID" /></td>.
      </tr>.
      <tr>.
        <td>Username</td>.
        <td><com:TLabel ID="Username" /></td>.
      </tr>.
      <tr>.
        <td>Employee Number</td>.
        <td><%= $this->Page->getProfile('emp_no')%></td>.
      </tr>.
      <tr>.
        <td>Title</td>.
        <td><com:TLabel ID="FacultyTitle" /></td>.
      </tr> .
```

```
function logout().
```

```
{.
```

```
    $acl = new UpsAcl();.
```

```
    $username = $acl->getUsername();.
```

```
    $acl->logoutUser();.
```

```
    .
```

```
    $this->redirect($acl->getLoginURL(false));.
```

```
}|.
```



```
function myprofile().
{.
  $acl =& $this->getAcl();.
  $id  = $acl->getUserID();.
  $this->models['user']->setId($id);.

  if (empty($this->params['data'])) .
  {.
    $this->params['data'] = $this->models['user']->read();.
  }.
  else .
  {.
    if ($this->models['user']->save($this->data)) .
    {.
      $this->flash(i18n::get('user_profile_updated'), '/users/myprofile');.
    } .
    else.
    {.
      $this->validateErrors($this->models['user']);.
    }.
  }.
}.
}.
```

```
<h1><?= i18n::get('my_profile') ?></h1>.
.
<div class="gen-form">.
<?= $html->formTag('myprofile') ?>.
.
  <dt><?= $form->labelTag('username', i18n::get('username')) ?></dt>.
  <dd><?= $html->input('user/username') ?></dd>.
  <?= $html->tagErrorMsg('user/username', i18n::get('username_required')); ?>.
.
  <dt><?= $form->labelTag('password', i18n::get('password')) ?></dt>.
  <dd><?= $html->password('user/password') ?></dd>.
  <?= $html->tagErrorMsg('user/password', i18n::get('password_required')) ?>.
.
  <dt><?= $form->labelTag('password_confirm', i18n::get('password_confirm')) ?></dt>.
  <dd><?= $html->password('user/password_confirm') ?></dd>.
  <?= $html->tagErrorMsg('user/password_confirm', i18n::get('password_must_match')) ?>.
.
  <dd><p><?= $html->submit(i18n::get('save')) ?></p></dd>.
</form>.
</div>.
```

# Rails



“Do one thing every day that scares you.”  
—Eleanor Roosevelt

# First Impressions

# Directory Structure

- ▼ expenses
  - ▼ app
    - ▼ controllers
      - application.rb
    - ▶ helpers
    - ▶ models
    - ▶ views
    - ▶ components
    - ▶ config
    - ▶ db
    - ▶ doc
    - ▶ lib
    - ▶ log
    - ▶ public
    - ▶ Rakefile
    - ▶ README
    - ▶ script
    - ▶ test
    - ▶ vendor

# Scaffold

“When one puts up a building one makes an elaborate scaffold to get everything into its proper place. But when one takes the scaffold down, the building must stand by itself with no trace of the means by which it was erected.”

- Andres Segovia

```
app/controllers/  
app/helpers/  
app/views/reviews  
app/views/layouts/  
test/functional/  
model  
  app/models/  
  test/unit/  
  test/fixtures/  
  app/models/review.rb  
  test/unit/review_test.rb  
  test/fixtures/reviews.yml  
app/views/reviews/_form.rhtml  
app/views/reviews/list.rhtml  
app/views/reviews/show.rhtml  
app/views/reviews/new.rhtml  
app/views/reviews/edit.rhtml  
app/controllers/reviews_controller.rb  
test/functional/reviews_controller_test.rb  
app/helpers/reviews_helper.rb  
app/views/layouts/reviews.rhtml  
public/stylesheets/scaffold.css
```



- **Create**
- **Read**
- **Update**
- **Delete**

```
class ReviewsController < ApplicationController.  
  def index.  
    list.  
    render :action => 'list'.  
  end.  
.  
  def list.  
    @review_pages, @reviews = paginate :reviews, :per_page => 10  
  end.  
.  
  def show.  
    @review = Review.find(params[:id]).  
  end.  
end.
```

```
def new.  
  @review = Review.new.  
end.  
  
def create.  
  @review = Review.new(params[:review]).  
  if @review.save.  
    flash[:notice] = 'Review was successfully created.'  
    redirect_to :action => 'list'.  
  else.  
    render :action => 'new'.  
  end.  
end.
```

```
def edit.  
  @review = Review.find(params[:id]).  
end.  
  
def update.  
  @review = Review.find(params[:id]).  
  if @review.update_attributes(params[:review]).  
    flash[:notice] = 'Review was successfully updated.'  
    redirect_to :action => 'show', :id => @review.  
  else.  
    render :action => 'edit'.  
  end.  
end.
```

```
def destroy.  
  Review.find(params[:id]).destroy.  
  redirect_to :action => 'list'.  
end.
```



# Listing reviews

Content Reviewer Created at Updated at

[New review](#)

# New review

Content

great food!

Reviewer

jack black

Created at

2006 September 26 09 : 42

Updated at

2006 September 26 09 : 42

Review was successfully created.

# Listing reviews

Content Reviewer	Created at	Updated at	
great food! jack black	Tue Sep 26 09:42:00 China Standard Time 2006	Tue Sep 26 09:42:38 China Standard Time 2006	<a href="#">Show</a> <a href="#">Edit</a> <a href="#">Destroy</a>

[New review](#)



**Content:** great food!  
**Reviewer:** jack black  
**Created at:** Tue Sep 26 09:42:00 China Standard Time 2006  
**Updated at:** Tue Sep 26 09:42:38 China Standard Time 2006  
[Edit](#) | [Back](#)



# Editing review

Content

great food!

Reviewer

jack black

Created at

2006 September 26 09 : 42

Updated at

2006 September 26 09 : 42

Done

Reviews: list - Mozilla Firefox

File Edit View Go Bookmarks Tools Help del.icio.us

http://localhost:3000/reviews/list

Morning ClickTheCity - TV G... Speakeasy - Speed ... ruby Blogs Programming Blogs FogBugz Upstrat - Dashboard Speedtest.net

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

## Listing reviews

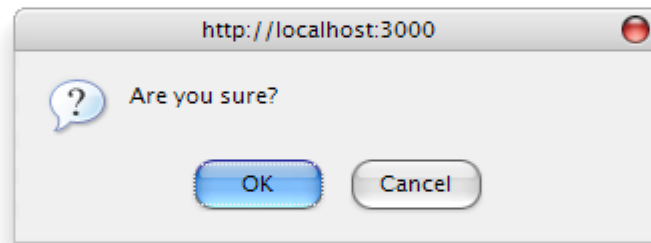
**Content Reviewer**

**Created at**

**Updated at**

great food! jack black Tue Sep 26 09:42:00 China Standard Time 2006 Tue Sep 26 09:42:38 China Standard Time 2006 [Show](#) [Edit](#) [Destroy](#)

[New review](#)



# Listing reviews

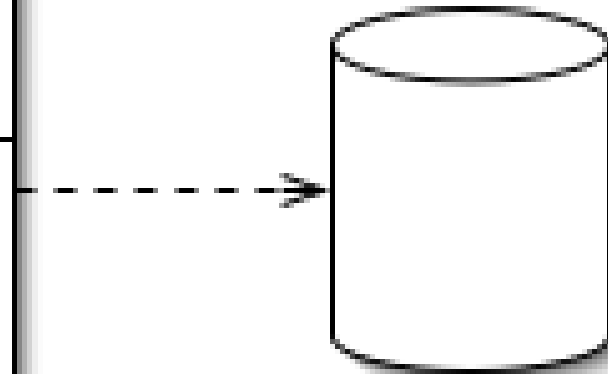
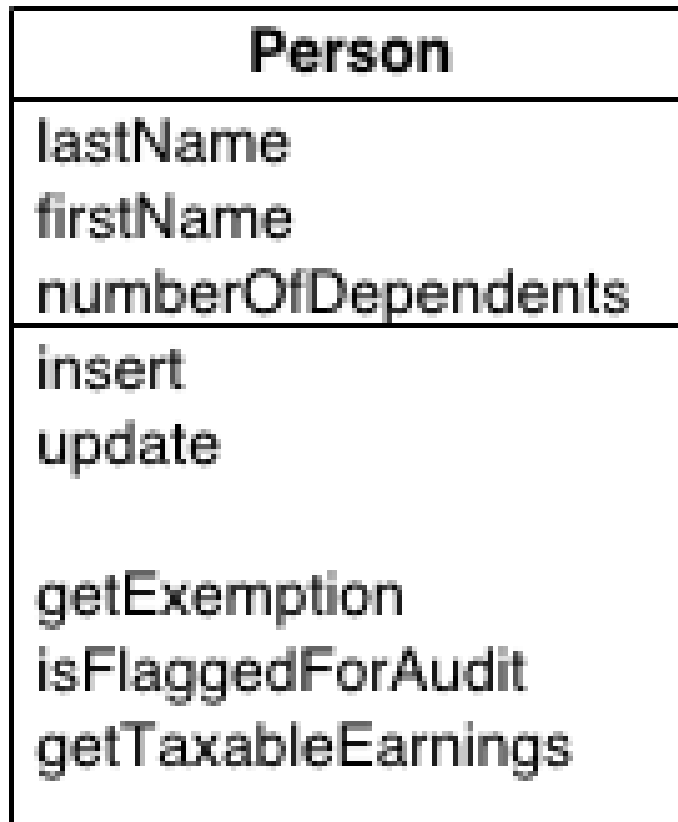
Content Reviewer Created at Updated at

[New review](#)

# Active Record

“We have a ‘strategic’ plan. It’s called doing things.”

—Herb Kelleher, founder, Southwest Airlines



Source: [www.martinfowler.com](http://www.martinfowler.com)

# The Article Model

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<b>id</b>	int(11)			No		auto_increment
<input type="checkbox"/>	<b>title</b>	varchar(255)	latin1_general_ci		Yes	NULL	
<input type="checkbox"/>	<b>body</b>	text	latin1_general_ci		Yes	NULL	

```
class Article < ActiveRecord::Base.  
end.
```

```
class Article < ActiveRecord::Base.  
end.  
  
Article.find(1) # SELECT * FROM articles WHERE id = 1.  
Article.find(:first) # SELECT * FROM articles LIMIT 0,1.  
Article.find(:all) # SELECT * FROM articles.  
  
article = Article.new.  
article.title = 'World Trade Center'.  
article.body = 'The world trade center is a movie starring Nicolas Cage....'.  
article.save.  
  
Article.find_by_title('World Trade Center') .  
  # SELECT * FROM articles WHERE title = 'World Trade Center'.
```



```
article.title # 'World Trade Center'.
```

```
article.body # 'The world trade center is a movie starring Nicolas Cage....'.
```

**SQL for Free**

```
Article.find(:all, :conditions => "title = 'My Article Title'").
# SELECT * FROM articles WHERE title = 'My Article Title'.
.
Article.find(:all, :order => 'body desc').
# SELECT * FROM articles ORDER BY body desc.
.
Article.find_all_by_title_and_body('my title', 'my body').
# SELECT * FROM articles WHERE title = 'my title' AND body = 'my body'.
.
Article.count(:conditions => "title = 'My title'").
# SELECT COUNT(*) as count_all FROM articles WHERE title = 'My title'.
.
Article.find(:all, :conditions => ["body LIKE ?", 'some search string here']).
# SELECT * FROM articles WHERE body LIKE 'some search string here'.
```

# Associations

- **Has One**
- **Belongs To**
- **Has Many**
- **Has and Belongs to Many**

# Author Model

Field	Type	Collation	Attributes	Null	Default	Extra
<u>id</u>	int(11)			No		auto_increment
name	varchar(255)	latin1_general_ci		Yes	NULL	

# Article Model

Field	Type	Collation	Attributes	Null	Default	Extra
<u>id</u>	int(11)			No		auto_increment
title	varchar(255)	latin1_general_ci		Yes	NULL	
body	text	latin1_general_ci		Yes	NULL	
author_id	int(11)			Yes	NULL	

```
class Article < ActiveRecord::Base
  belongs_to :author.
end.
.
.
class Author < ActiveRecord::Base.
  has_many :articles.
end.
```

```
.
author = Author.new(:name => 'John Doe').
.
author.articles << Article.new(:title => "My First Article", :body => "Hello everybody...").
author = Author.new(:name => 'Jhon Doe').
.
author.articles << Article.new(:title => 'My First Article',.
                               :body => "Hello everybody...").
.
author.save # author and article are saved .
           # in a single database transaction.
.
Author.find(:all, :include => :articles).
# SELECT * FROM authors .
# JOIN article ON articles.author_id = author.id.
```



```
author = Author.find(1) # SELECT * FROM author WHERE id = 1.
-
author.articles # SELECT * FROM articles WHERE articles.author_id = 1.
-
author.articles.create(:title => 'My Second Article',..
                       :body => "I'm back guys...").
# INSERT INTO articles (title, body, author_id).
# VALUES ('My Second Article', "I'm back guys...", 1).
-
article = Article.find(1) # SELECT * FROM articles WHERE id = 1.
-
article.author_id # returns 1.
-
article.author # SELECT * FROM authors WHERE id = 1.
```

**In Action**

- ```
class Patient < Person.  
  has_many :consultations.  
end.
```

- ```
class Consultation.  
  belongs_to :patient.  
end.
```

```
def create.  
  @consultation = Consultation.new.  
  @patient = Patient.find(params[:patient_id]).  
  @patient.consultations << @consultation if @consultation.new_record?.  
  @consultation.attributes = params[:consultation].  
  @consultation.save!.  
  flash[:notice] = 'Consultation was successfully saved.'.  
  redirect_to patient_path(@patient).  
end.  
.  
alias update create.  
.  
def destroy.  
  Consultation.find(params[:id]).destroy.  
  redirect_to patient_path(@patient).  
end.
```

```
class Patient < Person.  
  has_many :consultations.  
  
  .  
  validates_presence_of      :first_name, :last_name.  
  validates_numericality_of :age.  
  validates_uniqueness_of   :code.  
  validates_email_of       :email|.  
  
  .  
end.
```

```
class Consultation.  
  belongs_to :patient.  
  .  
  validates_associated :patient.  
  validates_presence_of :notes.  
end.
```

**Convention  
over  
Configuration**

# Smart Defaults



```
.  
class Article.  
  belongs_to :author.  
end.  
  
.  
class Author.  
  has_many :articles.  
end.
```

```
.
class Article.
  belongs_to :author, :class_name => 'Author', :foreign_key => 'author_id'.
end.

class Author.
  has_many :articles, :class_name => 'Article', :foreign_key => 'author_id'.
end.
```

```
class Article.  
  set_table_name 'articles'.  
  set_primary_key 'id'.  
end.  
  
.  
# /articles/show/1.  
# maps to ArticlesController#show, no routing configuration necessary.  
class ArticlesController < ApplicationController.  
  def show.  
  end.  
end.
```

# Plugins

“We eat change for breakfast!”

—Harry Quadracci, founder, QuadGraphics

# Some Examples

- **Acts as Versioned**
- **Acts as Sentry**
- **Acts as Dropdown**
- **Localization**

# Active Support

```
5.days.from_now.  
5.days.ago.  
  
.  
"article".pluralize.  
# => articles.  
  
.  
"mythical_creature".camelize.  
# => "MythicalCreature".  
  
.  
"mythical_creature".camelize.constantize.  
# => MythicalCreature.  
  
.  
Article.find.  
# will catch uninitialized constant "Article" exception.  
# eventually leading to the following command.  
# require 'app/models/article.rb' .
```

# Ruby



“Nothing is so contagious as enthusiasm.”

—Samuel Taylor Coleridge



# Closures

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”

- Martin Fowler

# Collection Closures

Taken from [www.martinfowler.com](http://www.martinfowler.com)

```
employees.each { |e| e.do_something }.  
.  
for e in employees.  
  e.do_something.  
end.  
.  
employees.each do |e|.  
  e.do_something.  
end.
```

```
managers = employees.select {|e| e.manager?}.  
.  
managers = [].  
for e in employees.  
  if e.manager?.  
    managers << e .  
  end.  
end.
```

```
offices = employees.collect {|e| e.office}.  
  
offices = [].  
for e in employees.  
  offices << e.office.  
end.
```

```
allManagers = employees.all? { |e| e.manager? }.
noManagers = ! employees.any? { |e| e.manager? }.
.
.
managers, plebs = employees.partition { |e| e.manager? }.
|
```

```
total = employees.inject(0) {|result, e| result + e.salary}.
```

```
.
```

```
total = 0.
```

```
for e in employees.
```

```
  total += e.salary.
```

```
end.
```

# Open Classes

```
class Fixnum.  
  def inches.  
    self.  
  end.  
  alias inch inches.  
  .  
  def feet.  
    self * 12.  
  end.  
  alias foot feet.  
  .  
  def yards.  
    feet * 3.  
  end.  
  alias yard yards.  
end.
```



1.inch # => 1.

2.inches # => 2.

1.foot # => 12.

2.feet # => 24.

1.yard # => 36.

10.yards # => 360.

.

10.yards + 1.inch + 2.feet # => 385.

```
module Localization.  
  attr_accessor :lang.  
  .  
  @@l10s = { :default => {} }.  
  @@lang = :default.  
  .  
  # the rest of the localization code....  
  .  
end.  
  
class Object.  
  def _(*args); Localization._(*args); end.  
end.
```

**Gems**

actionmailer (1.2.5, 1.2.3, 1.2.1)

Service layer for easy email delivery and testing.

actionpack (1.12.5, 1.12.3, 1.12.1)

Web-flow and rendering framework putting the VC in MVC.

actionwebservice (1.1.6, 1.1.4, 1.1.2)

Web service support for Action Pack.

activerecord (1.14.4, 1.14.3, 1.14.2)

Implements the ActiveRecord pattern for ORM.

activesupport (1.3.1)

Support and utility classes used by the Rails framework.

BlueCloth (1.0.0)

BlueCloth is a Ruby implementation of Markdown, a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).

capistrano (1.1.0)

Capistrano is a framework and utility for executing commands in parallel on multiple remote machines, via SSH. The primary goal is to simplify and automate the deployment of web applications.

**Rake**

```
rake db:migrate.
```

```
.
```

```
rake deploy.
```

```
.
```

```
rake db:test:clone_structure.
```

```
.
```

```
rake test:units.
```

```
.
```

```
rake test:functionals.
```

```
.
```

```
rake test:integration.
```

```
.
```

```
rake test.
```

# XP:

# Extreme

# Programming

“Never doubt that a small group of committed people can change the world. Indeed it is the only thing that ever has.”

—Margaret Mead

# What's it all about?

- “Ready...Fire...Aim”
- Like Driving
- Discipline

“We may not be interested in chaos but chaos is interested in us.”—Robert Cooper,



# Test Driven Development

```
def test_class_type.  
  assert_kind_of Todos::Task, tasks(:first_task).  
end.
```

```
def test_has_open_sub_tasks?.
  task = tasks(:first_task).
  task.sub_tasks << Todos::SubTask.create(:details => 'some details, uncompleted').
  .
  assert task.has_open_sub_tasks?.
end.
```

**Feedback =  
Quick Iterations**

# Continuous Integration

```
.....  
.....  
.....  
Finished in 19.783757 seconds.  
  
299 tests, 771 assertions, 0 failures, 0 errors  
Loaded suite /usr/bin/rake  
Started
```

# Metaprogramming

```
class ChangeObserver < ActiveRecord::Observer.  
  observe Problem::Concern, Problem::Analysis, Problem::Action, Todos::Task.  
.  
  track_changes_for :assignee do .  
    set :details, '%class_name% Assignee assigned to %assignee%'.  
    set :action_that_happened, '%class_name% updated by'.  
  end.  
.  
  track_changes_for :verifier do .  
    set :details, '%class_name% Verifier assigned to %verifier%'.  
    set :action_that_happened, '%class_name% updated by'.  
  end.  
.  
  track_changes_for :details do .  
    set :action_that_happened, '%class_name% updated by'.  
    set :details, subject.details.  
  end .
```



```
class Policy::Status::Rules < Policy::Status::Base.  
  policy(:accomplished) { status == ACCOMPLISHED or status == FAILED }.  
  policy(:completed)   { status == COMPLETED }.  
  policy(:not_completed) { status == NOT_COMPLETED }.  
  policy(:closed)      { status == CLOSED }.  
  policy(:valid)       { status == VALID }.  
  policy(:invalid)     { status == INVALID }.  
  policy(:verified)    { status == VERIFIED }.  
  policy(:rejected)    { status == REJECTED }.
```

```
class Components::One::MonitoringDefinitions < Components::One::Base.  
  define :target_companies_yr2.  
  define :companies_covered_grants_yr2.  
  define :percent_companies_covered_yr2.  
  define :percent_companies_more_than_200_workers, :as => proc { 'num_of_workers > 200' },  
    :after => proc { (result / total_companies) * 100 }.  
  define :total_companies_covered_more_than_200_workers.  
  define :yr2_companies_covered_more_than_200.  
  define :percent_workplace_fpmt_trained_grants_yr2.  
  define :percent_workplace_fp_policy_in_place_grants_yr2.  
  define :percent_workplace_fp_installed_grants_yr2.  
  define :total_workplace_fpmt_trained_grants_yr2.  
  define :total_workplace_fp_policy_in_place, :as => proc { 'has_fp = 1' }.  
  define :total_workplace_fp_installed.  
  define :total_companies, :as => :all, :visibility => :hidden.  
end.
```

**Domain  
Specific  
Languages**

# Ruby Smash!

A rip-off of Dwemthy's Array

```
class Bowser < Character.  
  life 1340      # tough scales.  
  strength 451  # bristling veins.  
  charisma 1020 # toothy smile.  
  weapon 939    # fire breath.  
end.
```

```
class Link < Character.  
  traits :bombs.  
.  
  life 10.  
  strength 2.  
  charisma 44.  
  weapon 4.  
  bombs 3.
```

```
# little boomerang.
def ^( enemy ).
  fight( enemy, 13 ).
end.
# the hero's sword is unlimited!!.
def /( enemy ).
  fight( enemy, rand( 4 + ( ( enemy.life % 10 ) ** 2 ) ) ).
end.
# lettuce will build your strength and extra ruffage.
# will fly in the face of your opponent!!.
def %( enemy ).
  lettuce = rand( charisma ).
  puts "[Healthy lettuce gives you #{ lettuce } life points!!]"
  @life += lettuce.
  fight( enemy, 0 ).
end.
# bombs, but you only have three!!.
def *( enemy ).
  if @bombs.zero?.
    puts "[UHN!! You're out of bombs!!]"
    return.
  end.
  @bombs -= 1.
  fight( enemy, 86 ).
end.
end
```

```
.  
class Mario < Character.  
  life 46.  
  strength 35.  
  charisma 91.  
  weapon 2.  
end
```

```
.  
link = Link.new.  
mario = Mario.new.
```

# Fight!

```
>> link ^ mario.  
[You hit with 2 points of damage!].  
#.  
[Your enemy hit with 28 points of damage!].  
☐Link has died.]
```



```
.
class Luigi < Creature.
  life 320.
  strength 6.
  charisma 144.
  weapon 50.
end.
.
class CaptainFalcon < Creature.
  life 655.
  strength 192.
  charisma 19.
  weapon 109.
end.
.
class SolidSnake < Creature.
  life 901.
  strength 560.
  charisma 422.
  weapon 105.
end.
.
class Bowser < Creature.
  life 1340      # tough scales.
  strength 451  # bristling veins.
  charisma 1020 # toothy smile.
  weapon 939   # fire breath.
end          |.
```

# That's it!

## Thanks for coming to the presentation

- Visit our website <http://www.upstrat.com>
- If you've got anymore questions, feel free to contact us
- Upstrat is seeking Ruby on Rails enterprising projects
- Email us at [amcvega@upstrat.com](mailto:amcvega@upstrat.com) and [cyx@upstrat.com](mailto:cyx@upstrat.com)