

Jan Berdajs

MrBrdo

www.mrbrdo.net

**Ruby on Rails
Prosojnice + material:
<http://zeta.mrbrdo.net/ror.zip>**

Outline

- Ruby - language
 - irb
 - modules, classes (constructor, .methods/instance_methods), methods, leaving out parenthesis
 - vars in strings
 - loops
 - symbols, loops, embedded ruby, blocks
- Ruby on Rails - framework

- Interactive Ruby
 - interactive console
 - testing
 - debugging
 - playing around

- Test
 - math, methods...
 - gems that improve irb: hirb (database outputs as tables...), wirble (tab-completion, history, coloring...)

Modules, classes, methods (...OOP)

- Modules, classes, methods
 - moduli.rb
 - constructor (initialize), no destructors (garbage collection), but it can be done (usually not needed/wanted)
 - leaving out parentheses and returns
 - single inheritance for classes, multiple inheritance for modules (MixIns...) similar to Java's Interfaces

- static methods
- constants - capital first letter, usually uppercase (or camel)
- attr_accessor for attributes (attr_writer, attr_reader)
- methods with question/exclamation marks in their names

misc

- variables in strings “bla #{spr}” double-quoted
- zanke.rb
 - while, until, for, break, next, redo (no condition checking), retry (for exceptions)
- .html.erb (<% %>, <%= %>) embedded ruby, templates
- blocks.rb: lambda funcs, procs, blocks (=closures in ruby), variable scope, passing
- symbols - immutable strings (unchangable, only 1 instance in memory - faster) - :symbol, :”bla bla simbol”, “bla bla string”.intern -> symbol

Ruby on Rails

- July 2004
- Framework for development of web applications
- A LOT of 'copies' (CakePHP, CodeIgniter, Django, Symfony, Zend framework...)
- MVC
 - Model : ActiveRecord - business logic (represents data e.g. in the database)
 - View : ActionView - display of data (e.g. HTML/CSS/Javascript)
 - Controller : ActionController - gets input data, uses the model and can generate data for the view, do a redirect etc... basically all non-model related code
- CoC - convention over configuration
- DRY - don't repeat yourself
- Scaffolding, WEBrick, environment console, migrations
- RESTful architecture - Representational state transfer ; resources, GET, PUT, POST, DELETE

Typical project

- rails appname -d mysql
- cd appname
- structure (important directories): config/, app/controllers, app/models, app/views

- config/database.yml
- rake db:create
- rm public/index.html
- config/routes.rb
- script/generate scaffold Post name:string title:string content:text
- rake db:migrate

Deployment

- People used to use Mongrel clusters
- Nowadays: **Apache** or Nginx with **mod_rails** (Passenger mod)
- Version control: GIT
- Fully automated deployment with Capistrano tasks and a GIT repository. When we update the project we can upload it, update the DB, restart the server, etc... all with one command (cap deploy). No need for FTP, logging on to the server yourself etc. - it's all automagical, uses GIT and SSH.

Scalability

- There's no real problem for small-to-medium projects! By the time you really need to scale well your project will probably be huge.
- Twitter had problems (but they had A LOT of traffic), they've now rewritten a **small** part of the **backend** in Scala, but the majority of the code is still written in RoR.
- The speed of script execution is one of the last things that we should go optimize. There's a lot of other things much more important for load times (merging CSS, JS scripts - only 1 extra word in RoR), caching etc... Ruby has most of this built-in.
- Ruby 1.9 is up to ~100 times faster than 1.8 (in some cases), and in some cases even faster than Python. It depends on the code (ActiveRecord still slow).
Eg. recursive fibonacci: R1.8: 158s, Python: 31s, R1.9: 11s
- Currently Ruby 1.8 is still recommended for RoR, because some native extensions don't exist for 1.9 yet.
- The state is constantly improving and should be a non-issue in a few years

Links

- <http://rubyonrails.org/documentation>
- <http://rubyonrails.org/screencasts>
- **<http://www.railscasts.com/> (the best)**
- <http://railslab.newrelic.com/scaling-rails> (for all sort of optimizations)
- http://grigio.org/pimp_my_gedit_was_textmate_linux
- <https://help.ubuntu.com/community/RubyOnRails>
- freenode: #RoR
- deployment: <http://www.capify.org/index.php/Capistrano>
- Passenger module (Apache, Nginx): <http://www.modrails.com/> + RubyEE (Enterprise Edition)
- Agile Web Development with Rails, Third Edition (best book on Ruby on Rails)
- Deployment:
 - Git: <http://www.urbanpuddle.com/articles/2008/07/11/installing-git-on-a-server-ubuntu-or-debian>
 - `gem install capistrano ; capify .` (where . is the project folder)
 - <http://github.com/guides/deploying-with-capistrano>
 - <http://www.zorched.net/2008/06/17/capistrano-deploy-with-git-and-passenger/>