

Ruby on Rails: A Lightning Fast Introduction

Daniel Piltch
dan@gradientblue.com

Ruby

- Yukihiro Matsumoto (Matz)
- Perl-ish (and Lisp-ish, and Smalltalk-ish)
- -- With a little Python thrown in
- Flexible
- Ducktyped

Ruby

- Concise and readable:
It makes programmers happy.
- Extensible through gem's
- Interpreted, not compiled

```
puts "Hello World!"
```

Everything, including a literal, is an object, so this works:

-199.abs # 199

"ruby is cool".length # 12

"Rick Astley".index("c") # 2

```
a = [1, 'hi', 3.14, 1, 2, [4, 5]]
```

```
p a[2]      # 3.14
```

```
p a.reverse # [[4, 5], 2, 1, 3.14, 'hi', 1]
```

```
p a.flatten.uniq # [1, 'hi', 3.14, 2, 4, 5]
```

```
my_hash = { :water => 'wet', :fire => 'hot' }  
puts my_hash[:fire] # Prints: hot
```

```
my_hash.each do |key, value|  
  puts "#{key} is #{value}"  
end
```

```
# Prints: water is wet  
#         fire is hot
```

Rails

- Framework based on MVC architecture
- DHH: David Heinemeier Hanson, Basecamp
- First released July 2004 as open source
- Convention over configuration
- Many! plugins available

Rails

- Includes prototype & scriptaculous
- Built-in 'scaffold' for creating basic CRUD application (more on this later)
- Integration with Java environment
 - -- jRuby - Java implementation of Ruby language
 - -- can drop a rails app packaged into a .war file onto an existing app server

```
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
```

```
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
```

```
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
```

```
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
```

your_new_rails_project/

.... /app

..... /controller

..... /helpers

..... /models

..... /views

..... /layouts

.... /components

.... /config

.... /db

.... /doc

.... /lib

.... /log

.... /public

.... /script

.... /test

.... /tmp

.... /vendor

README

Rakefile

IDEs

- TextMate (Mac Only)
- Aptana RadRails (Eclipse)
- NetBeans
- Roll your own (gEdit, jEdit, etc.)