

Microsoft®
tech·days
2012

palais des
congrès
Paris

7, 8 et 9
février 2012

Introduction à Ruby

Mardi 7 février
Nicolas Ledez
Architecte système
Orange Business Services
It&L@bs



Nicolas Ledez

Nicolas Ledez



Nicolas Ledez



Nicolas Ledez

Microsoft
tech·days
2012



lifecycle



IT & L@bs



Nicolas Ledez

Microsoft
tech·days
2012



Nicolas Ledez



Rennes on Rails

Happy coding!



Simon Courtois

Simon Courtois



CTO chez Selectra

Et maintenant Ruby

Ruby : l'histoire



- ➡ Créée en 1995
- ➡ Inspiré de Smalltalk, Lisp, Eiffel, Ada, etc
- ➡ Japonais Yukihiro Matsumoto « matz »
- ➡ Index TIOBE
 - ▶ 12 2012
 - ▶ 10 2011

- ➡ Toyota Production System
- ➡ Gestion sans gaspillage

Oh! Une luciole
je voulais crier : « Regarde! »
mais j'étais seul
Taïg

Ruby 1/3



- ➡ Interprété
- ➡ Objet
- ➡ Multiparadigme
- ➡ Multiplateforme
- ➡ Libre, gratuit, etc

- ➞ Ramasse-miettes
- ➞ Gestion d'exceptions
- ➞ Modification du code en « live »
- ➞ Expressions rationnelles (Regexp)
- ➞ Blocs

- ➡ Héritage simple
- ➡ Mixin -> « héritage multiple »
- ➡ Extensions en C
- ➡ Les threads indépendants de l'OS
- ➡ Réflexion

Virtual Machine 1/2



➡Matz's Ruby Interpreter – Cruby

➡JRuby

➡IronRuby

➡Rubinius

➡MacRuby

Virtual Machine 2/2



- ➡ YARV (Yet another Ruby VM)
- ➡ XRuby - rb -> Java bytecode
- ➡ RubyJS - rb -> Javascript
- ➡ HotRuby

On commence ?

Les variables



➔ `var` -> variable locale

➔ `@var` -> variable d'instance

➔ `@@var` -> variable de classe

➔ `$var` -> variable globale

➔ `Var` -> constante

➔ etc.

Le langage / Procédural 1



```
def une_fonction  
  puts "Salut tout le monde !"  
end
```

```
une_fonction
```

Salut tout le monde !

Le langage / Procédural 2



```
def une_fonction(message)
  puts message
end
```

```
une_fonction "It's alive !"
```

It's alive !

Le langage / Objet



```
class UneClasse
  attr_accessor :message

  def initialize(message)
    @message = message
  end

  def afficher_message
    puts @message
  end
end
```

```
mess1 = UneClasse.new "Salut
tout le monde !"

mess1.afficher_message
mess1.message = "Au revoir"
puts mess1.message
```

Salut tout le monde !
Au revoir

Le langage / Tableau - bases

```
Array.new(5, "A")
```

```
a[0]['cat'] = 'feline'
```

```
a[1, 2]
```

```
a[1..3]
```

```
a[-3, 3]
```

Le langage / Tableau - cool



[1, 2, 3] + [4, 5]

#=> [1, 2, 3, 4, 5]

[1, 1, 3, 5] & [1, 2, 3]

#=> [1, 3]

[1, 2, 3] * 3

#=> [1, 2, 3, 1, 2, 3, 1, 2, 3]

[1, 1, 2, 2, 3, 3, 4, 5] - [1, 2, 4]

#=> [3, 3, 5]

Le langage / Dictionnaires

```
h = Hash.new("Go Fish")
```

```
h["a"] = 100
```

```
h["b"] = 200
```

```
h["a"] #=> 100
```

```
h["c"] #=> "Go Fish"
```

Et on pourrait y passer des heures

Classe / héritage



```
class MyArray < Array
  def clean!
    self.each do |e|
      self.delete(e) if e % 2
    end
  end
end
```

```
puts list.join " "
puts list.clean!.join " "
puts list.join " "
```

```
list = MyArray.new([ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ])
```

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10

2 4 6 8 10

Classe / Monkey 1

```
Class Array
  def clean!
    self.each do |e|
      self.delete(e) if e % 2
    end
  end
end
```

```
list = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
```

```
puts list.join " "
puts list.clean!.join " "
puts list.join " "
```

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10

2 4 6 8 10

Classe / Monkey 2 - map



```
Class Array
```

```
  def double
```

```
    self.map { |e| e * 2 }
```

```
  end
```

```
end
```

```
puts list.double.join " "  
puts list.join " "
```

```
list = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
```

```
2 4 6 8 10 12 14 16 18 20
```

```
1 2 3 4 5 6 7 8 9 10
```

Classe / Monkey 3 - reduce

```
Class Array
```

```
  def clean
```

```
    self.reduce([]) do |a, e|  
      a << e unless ((e % 2) == 1)
```

```
      a
```

```
    end
```

```
  end
```

```
end
```

```
puts list. clean.join " "  
puts list.join " "
```

```
list = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
```

2 4 6 8 10

1 2 3 4 5 6 7 8 9 10

Classe / Mixin



```
Module MyClean
  def clean!
    self.each do |e|
      self.delete(e) if e % 2
    end
  end
end

def double
  self.map { |e| e * 2 }
end

End
```

```
class Array
  include MyClean
end
```

```
class Hash
  include MyClean
end
```

```
Array.new.double
Hash.new.double
```

Autour du langage

Autour de Ruby / Dev



➡ Rspec, Cucumber, Minitest, ...

➡ Spork, Guard, ...

➡ Bundler, RVM, Rbenv, Pik, ...

➡ <http://www.bonjourgem.com/>

➡ +33,500 RubyGems.org

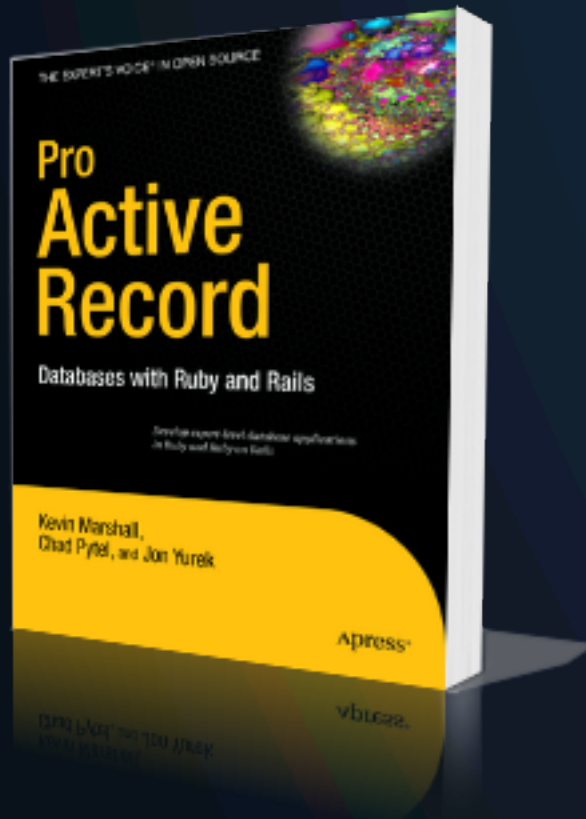
Autour de Ruby / Prod



- ➡ Rails, Sinatra, ...
- ➡ Spork, Guard, ...
- ➡ HAML, SASS, Compass, ...
- ➡ Capistrano, Puppetts, Cheff, ...
- ➡ 15% Ruby Github (2eme place)

ORM

Microsoft
tech·days
2012



DataMapper

Ruby Object Relational Mapper

ORM / Active record 1



```
class CreateTickets < ActiveRecord::Migration
  def change
    create_table :tickets do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end
end
```

ORM / Active record 2



```
class Ticket < ActiveRecord::Base
  validates_presence_of :name
  validates_presence_of :status
  belongs_to :status
end
```

ORM / DataMapper



```
class Lineltem
```

```
  include DataMapper::Resource
```

```
  property :order_id, Integer, :key => true
```

```
  property :item_number, Integer, :key => true
```

```
end
```

Framework Web



Sinatra

Rails



Uglifier

Rack

ERB

json

ActiveRecord

i18n

Demo Rails Par Simon Courtois

Sinatra



```
require 'sinatra'
```

```
get '/hi' do
```

```
  "Hello World!"
```

```
end
```

Templates



ERB

Templates / ERB

```
<p id="notice"><%= notice %></p>
```

```
<p>
```

```
  <b>Name:</b>
```

```
  <%= @region.name %>
```

```
</p>
```

```
<%= link_to 'Edit', edit_region_path(@region) %> |
```

```
<%= link_to 'Back', regions_path %>
```

Templates / HAML



```
%p#notice= notice
```

```
%p
```

```
  %b Name:
```

```
  = @region.name
```

```
= link_to 'Edit', edit_region_path(@region)
```

```
|
```

```
\#{link_to 'Back', regions_path}
```

J'ai Windows, je fais comment

Rails Installer

Microsoft
tech·days
2012



Project by  Engine Yard

Ready to deploy your app? [Start a free trial at Engine Yard](#)

RAILSINSTALLER GET UP & RUNNING WITH RAILS

STEP 1

DOWNLOAD
the KIT

For Windows only.

Mac and Linux versions *coming soon*.

Rails/Installer has everything you need to hit the ground running. In one easy-to-use installer, you get all the common packages needed for a full Rails stack. Download it now and be writing (and running) Rails code in no time. Packages included are:

- [Ruby](#) 1.9.2-p290
- [Rails](#) 3.1.1
- [Bundler](#) 1.0.18
- [Git](#) 1.7.6
- [Sqlite](#) 3.7.3
- [TinyTDS](#) 0.4.5
- [SQL Server support](#) 3.3.3
- [DevKit](#)

STEP 2

WATCH
the VIDEO



In this short video, you'll see the simple installation process and how to confirm that Ruby, Rails, Git, etc are all installed properly. Watch it [here](#) or if you have trouble, view it on [Vimeo](#).

STEP 3

WHAT'S
NEXT?

Have troubles installing?

[Check the Rails Installer Google Group.](#)

You've got everything installed. You've written (and ran) some code. If you find yourself asking *where do I go from here*, don't worry. You're not alone.

The answer is *just about anywhere*, but here's a handful of good places to start. The official [Ruby on Rails site](#), [blog](#), [Twitter account \(@rails\)](#), and [mailing list](#). If you're into IRC, swing by [#rubyonrails](#) on [freenode.net](#). There's always the ever-growing [Ruby on Rails guides](#). If you learn by watching, there are more [screencasts](#) than you can shake a stick at. Some are free, some cost money. Envy Labs produces the [Ruby 5 News Show](#), [Sby5](#) produces podcasts like [The Dev Show](#) and [The Ruby Show](#).

Rails Installer / DevKit



github



[Explore](#) [Gist](#) [Blog](#) [Help](#)



nledez

6



👤 oneclick / [rubyinstaller](#)

👁 Watch

🍴 Fork

👁 342

🍴 30

Code

Network

Pull Requests 2

Issues 15

Wiki 13

Stats & Graphs

[Home](#)

[Pages](#)

[Wiki History](#)

[Git Access](#)

Development Kit

New Page

Edit Page

Page History

DevKit Overview

The DevKit is a toolkit that makes it easy to build and use native C/C++ extensions such as RDiscount and RedCloth for Ruby on Windows.

Simply download, double-click, choose an installation directory, run the Ruby install helper script, possibly tweak a `config.yml` file for your system specifics, and you're ready to start using native Ruby extensions.



The **Ruby** Programming Language
on the JVM



Download the latest release - **1.6.6**

[tar](#) | [zip](#) | [dmg](#) | [exe](#) | [exe\(x64\)](#) | [exe+jre](#) | [exe+jre\(x64\)](#)

WHY JRUBY?

GET STARTED

LEARN

GET INVOLVED

The Best of the JVM



- High performance
- Real threading
- Vast array of libraries

It's Just Ruby



- Ruby 1.8.7 & 1.9.2 compatible
- Both versions in single install

Platform Independent



- Easy to install
- Easy migration
- No hassles

IronRuby

Microsoft
tech·days
2012



IronRuby the Ruby programming language for the .NET Framework

IronRuby is an open-source implementation of the Ruby programming language which is tightly integrated with the .NET Framework. IronRuby can use the .NET Framework and Ruby libraries, and other .NET languages can use Ruby code just as easily.

```
# namespaces are modules
include System::Collections::Generic

# indexers constrains type
d = Dictionary[String, Fixnum].new

# Ruby idioms just work
d['Hello'] = 1
d['Hi'] = 2

# this gives a TypeError
d[3] = 3

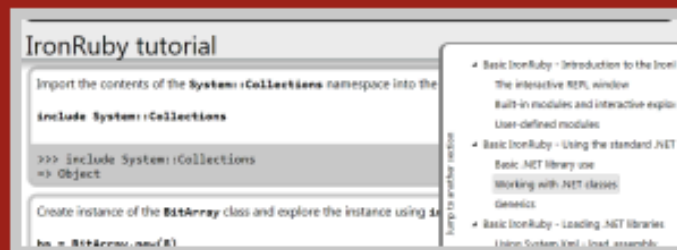
# Enumerable methods work
d.each{|kvp| puts kvp}
```

Download
IronRuby
1.1

1.1.3 released on 2011-3-13
[release notes](#) | [source](#)

Try Ruby in the browser

Quickly run Ruby code in your browser, without installing IronRuby.



Experience a more interactive
.NET and Ruby development
experience with Ruby Tools for
Visual Studio.

IronRuby



WebApp10 - Microsoft Visual Studio

File Edit View Glt Project Build Debug Team Data Tools .NET Reflector Architecture Test Analyze Window Help

posts_controller.rb x post.rb 20100419080331_create_posts.rb application_controller.rb routes.rb

PostsController

```
class PostsController < ApplicationController
  # GET /posts
  # GET /posts.xml
  def index
    @posts = Post.all

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @posts }
    end
  end
end
```

100 %

IronRuby Interactive

```
* Post
=> Post(id: integer, title: string, body: text, published: boolean,
created_at: datetime, updated_at: datetime)
* Post.all
=> []
* Post.create :title => 'Creating Post', :body => "this is a post"
=> #<Post id: 1, title: "Creating Post", body: "this is a post",
published: nil, created_at: "2010-04-19 08:10:08", updated_at:
"2010-04-19 08:10:08">
* Post.count
=> 1
```

Solution Explorer

Solution 'WebApp10' (1 project)

- WebApp10
 - References
 - Search Path
 - app
 - controllers
 - application_contro
 - posts_controller.rb
 - helpers
 - models
 - post.rb
 - views
 - layouts
 - posts
 - config
 - environments
 - initializers
 - locales
 - boot.rb
 - database.yml
 - environment.rb
 - routes.rb
 - db
 - doc
 - lib
 - log
 - public
 - script
 - test
 - tmp
 - vendor

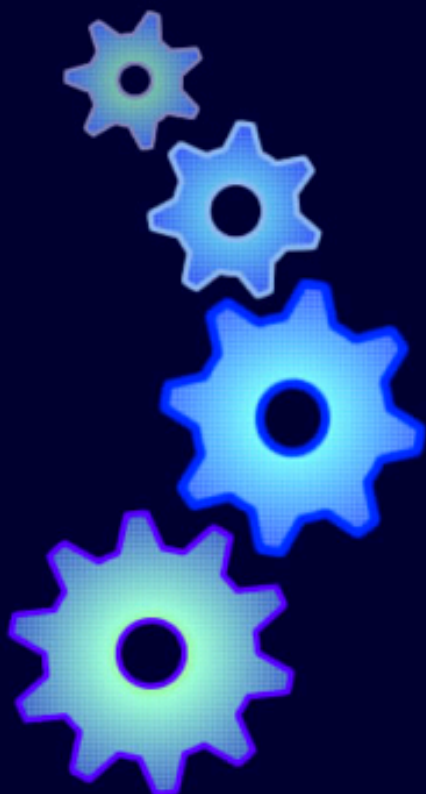
Solution Explorer Team Explorer

Error List Output Find Results 1 Find Symbol Results Pending Changes

Ready Ln 9 Col 44 Ch 44 INS



Visual Studio IDEs for professional programmers



amethyst^{1.5}

the flash platform in visual studio

ruby in steel^{2.0}

the professional ruby & rails ide



amethyst

[Product Page](#)

[Download Trial](#)

[Feature List](#)

[Tutorials](#)

ruby in steel

[Product Page](#)

[Download Trial](#)

[Feature List](#)

[Tutorials](#)

On the Blog...

[ActionScript Bubble Debugging – Amethyst 2 preview](#)

RubySteel

Microsoft
tech·days
2012



LittleBookOfRuby (Debugging) - Microsoft Visual Studio (Administrator)

File Edit View Project Build Debug Team Data Tools Architecture Test Analyze RubyInSteel Window Help

Debug AnyCPU error

data_save.rb x dir_array.rb LittleBookOfRuby each_loop.rb multi_array.rb adventure2.rb

CD showData()

```
104 # loads data from disk and recreates the array of
105 # cd objects, $cd_arr, from the data
106 def loadDB
107   input_data = File.read( $fn )
108   $cd_arr = YAML.load( input_data )
109 end
110
111 # prints the data from the array to screen in human-
112 # readable (YAML) format
113 def showData
114   puts( $cd_arr )
115 end
116
117 # Start of P
118
119
120 if File.exist?
```

Debugger View: \$cd_arr [0] <0x417d294>, [1] <0x417b9bc>, [2] <0x417970c>, [3] <0x4178000>

Debugger View: @numtracks: 12, @name: "Fab Grooves", @artist: "The Groovettes"

Ruby Explorer

- Array
- Bignum
- Binding
- Class
 - allocate
 - inherited
 - new
 - superclass
- Comparable
- ConditionVariable
- Data
- Dir
- Enumerable
- Errno
- Exception
- FalseClass
- FileTest
- File

Solution Explorer

- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- ed_db.yml
- data_save.rb

Properties

data_save.rb File Properties

Full Path C:\How\program

URL C:\How\program

Rails

Editor type Default

Layout File

Ruby

Arguments

Flags

Arguments

Arguments for the Ruby command

Ruby Console

```
name: Fab Grooves
numtracks: 12
~ !ruby/object:PopCD
artist: The Groovy Bees
genre: Rock
name: Greater Hits
numtracks: 12
Create (P)op CD (C)lassical CD, (
> >
```

Call Stack

Name	Lang
showData at line 114	Ruby
(main) at line 140	Ruby

Watch 1

Name	Value	Type
\$cd_arr	[...]	Array
[0]	<0x417d294>	CD
@numtracks	20	Fixnum
@name	"Greatest Hits"	String
@artist	"Dolly Parton"	String

Watch 1 Locals

Ready

RubySteel

The screenshot shows a RubySteel debugger interface. A 'Breakpoint Condition' dialog box is open, displaying the condition `$Testval == "Stop" && x == 100 && i > 6`. The 'Is true' radio button is selected. Below the dialog, the source code is visible, showing a `puts( i )` statement. A 'Watch 1' window at the bottom displays the current values of variables `$Testval`, `x`, and `i`.

Breakpoint Condition

When the breakpoint location is reached, the expression is evaluated and the breakpoint is hit only if the expression is true or has changed.

☒ Condition:

`$Testval == "Stop" && x == 100 && i > 6`

☒ Is true
☐ Has changed

OK Cancel

78 | i |
79 | puts(i)
80 |

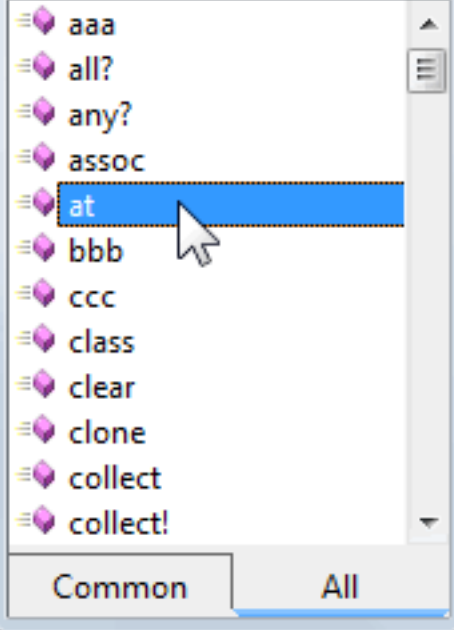
Watch 1

Name	Value	Type
\$Testval	"Stop"	String
x	100	Fixnum
i	7	Fixnum

RubySteel



```
72 class MyArray < Array
73   def aaa
74   end
75   def bbb
76   end
77   def ccc
78   end
79 end
80
81 ob = MyArray.new
82 ob.
83
```



array.at(index) -> obj or nil

Returns the element at index. A negative index counts from the end of self. Returns nil if the index is out of range. See also Array#[].
(Array#at is slightly faster than Array#[], as it does not accept ranges and so on.)

```
a = [ "a", "b", "c", "d", "e" ]
a.at(0)   #=> "a"
a.at(-1)  #=> "e"
```

Les ressources

Essayer maintenant



- ➡ <http://ironruby.net/try/>
- ➡ <http://tryruby.org/>
- ➡ <http://rubymonk.com/>

- ➡ <http://railsforzombies.org/>

Pour apprendre



- ➡ <http://ruby.railstutorial.org/>
- ➡ <http://railscasts.com/>

A suivre



- ➡ <http://nicolas.ledez.net/> @nledez
- ➡ Github : simonc @happynoff
- ➡ <http://www.camilleroux.com/> @CamilleRoux
- ➡ <http://matthieusegret.com/> @MatthieuSegret

La communauté



- ➡ <http://rubylive.fr/> @RubyLiveFR
- ➡ <http://www.rubyfrance.org/>
- ➡ <http://www.railsfrance.org/>
- ➡ Google groups:
- ➡ Rennes on Rails
- ➡ Apéros Ruby
- ➡ Railsfrance
- ➡ Ruby on Rails: Core
- ➡ @RubyJobsFR

Conclusion

