

RUBY 1.9



Guillaume Desrat

[<guillaume.desrat@rubyfrance.org>](mailto:guillaume.desrat@rubyfrance.org)

lundi 1er décembre 2008

Conference.first.orateur

Guillaume Desrat

guillaume.desrat@rubyfrance.org

RUBY 1.9

Ruby France

Portail de la communauté Ruby en France

<http://rubyfrance.org/>

Listes de discussion

<http://groups.google.com/group/rubyfr-public/>

Association loi de 1901

<http://rubyfrance.org/association/>

RUBY 1.9

une nouvelle
version ?

RUBY 1.9

une nouvelle
version...

incompatible !

RUBY 1.9

index dans les chaînes de caractères

```
"ruby on rails"[1]
```

```
# Ruby 1.8
```

```
=> 117
```

```
# Ruby 1.9
```

```
=> "u"
```

chaînes d'un caractère

?a

Ruby 1.8

=> 97

Ruby 1.9

=> "a"

variables de blocs

```
i = 0  
(1..5).each { |i| i }  
i
```

Ruby 1.8

=> 5

Ruby 1.9

=> 0

définition de Hash sans =>

```
{:action, "login", :controller, "users"}
```

```
# Ruby 1.8
```

```
=> {:action => "login", :controller => "users"}
```

```
# Ruby 1.9
```

```
=> SyntaxError
```

RUBY 1.9

ordre des clés dans les objets Hash

```
{:action => "show", :id => "1"}
```

Ruby 1.8

```
=> {:id => "1", :action => "show"}
```

Ruby 1.9

```
=> {:action => "show", :id => "1"}
```

Hash#select

```
{:action => "show", :id => "1"}.select do |k, v|  
  # ...  
end
```

Ruby 1.8

```
=> [[:action, "show"], [:id, "1"]]
```

Ruby 1.9

```
=> {:action => "show", :id => "1"}
```

Hash#select (suite)

```
{:action => "show", :id => "1"}.select do |x|  
  puts x  
end
```

*# Ruby 1.8 génère des avertissements
et affiche :*

```
[:action, "show"]  
[:id, "1"]
```

Hash#select (suite)

```
{:action => "show", :id => "1"}.select do |x|  
  puts x  
end
```

```
# Ruby 1.9 ne génère PAS d'avertissements  
# et affiche :  
"action"  
"id"
```

Hash#to_s

```
{:action => "login", :controller, "users"}.to_s
```

Ruby 1.8

```
=> "controllerusersactionlogin"
```

Ruby 1.9

```
=> "{:action=>\"login\", :controller=>\"users\"}"
```

Array#to_s

```
[:login, :logout].to_s
```

Ruby 1.8

`=> "loginlogout"`

Ruby 1.9

`=> "[:login, :logout]"`

case ... when avec des ":"

```
case errors[:password]  
  when "is too short" : require_more_chars  
  when "is too long"  : require_less_chars  
end
```

Ruby 1.8

=> nil

Ruby 1.9

=> SyntaxError

RUBY 1.9

mais encore ?

Hash#index n'est plus à utiliser

Fixnum#to_sym et String.each n'existent plus

\$KCODE est ignorée ; produit des avertissements

p x renvoie maintenant x, et non plus nil

instance_methods renvoie des symboles

...

syntaxe :

quoi de neuf ?

RUBY 1.9

Hash

```
{:id => @user.id, :class => "user"}
```

peut maintenant s'écrire :

```
{id: @user.id, class: "user"}
```

```
render action: "edit"
```

on peut mélanger les deux syntaxes

```
redirect_to controller: "users", :action => "login"
```

définition de blocs

```
p = lambda { |a, b, c| a + b + c }
```

peut maintenant s'écrire :

```
p = ->(a, b, c) { a + b + c }
```

exécution de blocs

```
p.call(6, 7, :*)
```

```
# peut maintenant s'écrire :
```

```
p.(6, 7, :*)
```

```
# ou :
```

```
p[6, 7, :*]
```

de nouvelles
fonctionnalités

RUBY 1.9

Paramètres des méthodes

```
require "date"
```

```
def voyage(depart = Date.today, destination)  
  puts "Départ le #{depart} pour #{destination}"  
end
```

```
voyage "Paris"
```

```
=> Départ le 2008-12-01 pour Paris"
```

Paramètres des méthodes (suite)

```
def calcule(*nombres, op)
  nombres.inject do |resultat, x|
    resultat = resultat.send(op, x)
  end
end
```

```
calcule(1, 2, 5, :+)
=> 8
```

```
calcule(6, 7, :*)
=> 42
```

Enumerable

```
arr = [1, 2].to_enum  
=> #<Enumerator:0x7e799e14>
```

```
arr.next  
=> 1
```

```
arr.next  
=> 2
```

```
arr.next  
=> StopIteration ; la séquence est réinitialisée
```

Enumerable

```
alphabet = ("a".."z").each  
=> #<Enumerator:0x7e799e14>
```

```
alphabet.next  
=> "a"
```

```
alphabet.next  
=> "b"
```

```
alphabet.rewind  
=> "a"
```

Enumerable

```
("a".."z").each.with_index do |lettre, ordre|  
  puts "#{ordre + 1} => #{lettre}"  
end
```

```
=> "1 => a"
```

```
    "2 => b"
```

```
    "3 => c"
```

```
    "4 => d"
```

```
    "5 => e"
```

```
    ...
```

Multilingualization (m17n)

```
# fichier "m17n_test.rb"
```

```
# coding: utf-8
```

```
puts "épée contient #{"épée".length} lettres"
```

```
% ruby m17n_test.rb  
épée contient 4 lettres
```

Multilingualization (m17n) (suite)

```
texte = File.read "m17n_test.rb"
```

```
=> "# coding: utf-8\n\nputs \"\xc3\xa9p\xc3\n\n\xe contient #{\n\n\".length} lettres\""
```

```
texte.encoding
```

```
=> #<Encoding:US-ASCII">
```

Multilingualization (m17n) (suite)

```
texte = File.read "m17n_test.rb", encoding: "utf-8"  
=> "# coding: utf-8\n\nputs \"épée contient \#{  
  \"épée\".length} lettres\""
```

```
texte.encoding  
=> #<Encoding:UTF-8">
```

Fiber

```
f = Fiber.new do
  i = 0
  loop do
    Fiber.yield i
    i += 1
  end
end
=> #<Fiber:0x8b7ce9b8>
```

```
loop { puts f.resume }
=> 0
=> 1
...
```

Fiber (suite)

```
nombre_premiers = Fiber.new do
  trouves = [2]
  loop do
    Fiber.yield trouves.last
    candidat = trouves.last + 1
    trouves << loop do
      break candidat unless trouves.any? do |i|
        candidat % i == 0
      end
      candidat += 1
    end
  end
end
```

Fiber (suite)

```
10.times { puts nombres_premiers.resume}
```

```
=> 2
```

```
3
```

```
5
```

```
7
```

```
11
```

```
13
```

```
17
```

```
19
```

```
23
```

```
29
```

Regexp

```
email = "guillaume.desrat@rubyfrance.org"
```

```
res = email.match /(?<user>\S+)@(?<host>\S+)/
```

```
res[:user]
```

```
=> "guillaume.desrat"
```

```
res[:host]
```

```
=> "rubyfrance.org"
```

mais encore ?

BasicObject est introduite au-dessus d'Object

Kernel#tap permet d'exécuter un bloc sur l'objet

! et != sont des méthodes

les symboles se comportent (un peu plus) comme des chaînes

Thread utilise les threads natifs

...

RUBY 1.9

quid des

performances ?

RUBY 1.9

Les machines de test

Mac Book Pro 2.2 GHz Intel Core 2 Duo

Mac OS X 10.4.11 (Tiger)

Parallels Desktop for Mac

2 machines virtuelles OpenBSD 4.4-release

Ruby 1.8.6-p287

Ruby 1.9.1-preview1

RUBY 1.9

La suite de Fibonacci

```
def fib(n)
  if n == 0 || n == 1
    n
  else
    fib(n - 1) + fib(n - 2)
  end
end

36.times do |i|
  puts "n = #{i} => #{fib(i)}"
end
```

La suite de Fibonacci (suite)

Ruby 1.8 : 52 secondes

Ruby 1.9 : 12 secondes

RUBY 1.9

ça va 4,3x

plus vite !

RUBY 1.9

Date

```
# bench_date.rb  
1000000.times do  
  Date.today + 365  
end
```

Ruby 1.8 : 102 secondes

Ruby 1.9 : 21 secondes

ça va 4,8x

plus vite !!!

RUBY 1.9

soyons
sérieux...

RUBY 1.9

ça va

globalement

plus vite

RUBY 1.9

“dois-je passer
à la 1.9 ?”

RUBY 1.9

“puis-je passer
à la 1.9 ?”

RUBY 1.9

incompatibilités

“je ne code pas comme ça, moi !”

RUBY 1.9

incompatibilités

(et les autres ?)

RUBY 1.9

Licence de ce document

vous êtes libres de **reproduire**, **distribuer** et **communiquer** cette création au public, selon les **conditions suivantes** :

paternité

vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'oeuvre)

pas d'utilisation commerciale

vous n'avez pas le droit d'utiliser cette création à des fins commerciales

pas de modification

vous n'avez pas le droit de modifier, de transformer ou d'adapter cette création