# RUBY
# DESDE CERO

# ¡Bienvenido!


## Julio García Pérez

Desarrollador - ElevenPaths (Telefónica)

@julioGarPer

# ¿Qué vamos a ver?

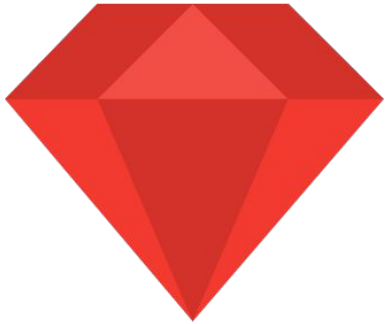
¡Conoceremos **Ruby** a fondo!

- Instalación
- Sintaxis
- Tipos de datos **Built-in**
- Estructuras de **control**
- POO
- Librerías (gemas)

# ¿Qué es Ruby ?

“A programmer best friend!”

- Creado en **Japón** por Yukihiro Matsumoto
- Lenguaje de propósito **general**
- Interpretado
- De **servidor**
- Dinámico
- Open Source
- Multiplataforma



# 100% orientado a Objetos

- **Everything** is an object!
  - Propiedades (**variables de instancia**)
  - Acciones (**métodos**)
  - Incluso los números!

```
5.times { print "We *love* Ruby -- it's outrageous!" }
```

# Muy **Simple**!

- ¡Pero **complejo** por dentro!
- **Claro** & fácil **sintaxis**

```ruby
# The Greeter class
class Greeter
  def initialize(name)
    @name = name.capitalize
  end

  def salute
    puts "Hello #{@name}!"
  end
end

# Create a new object
g = Greeter.new("world")

# Output "Hello World!"
g.salute
```

# Flexible

- Para programadores **responsables**
- Permite **modificar** su core
- ¡Sin restricciones!

```
class Numeric
  def plus(x)
    self.+(x)
  end
end

y = 5.plus 6
# y is now equal to 11
```

# Let’s Ruby !

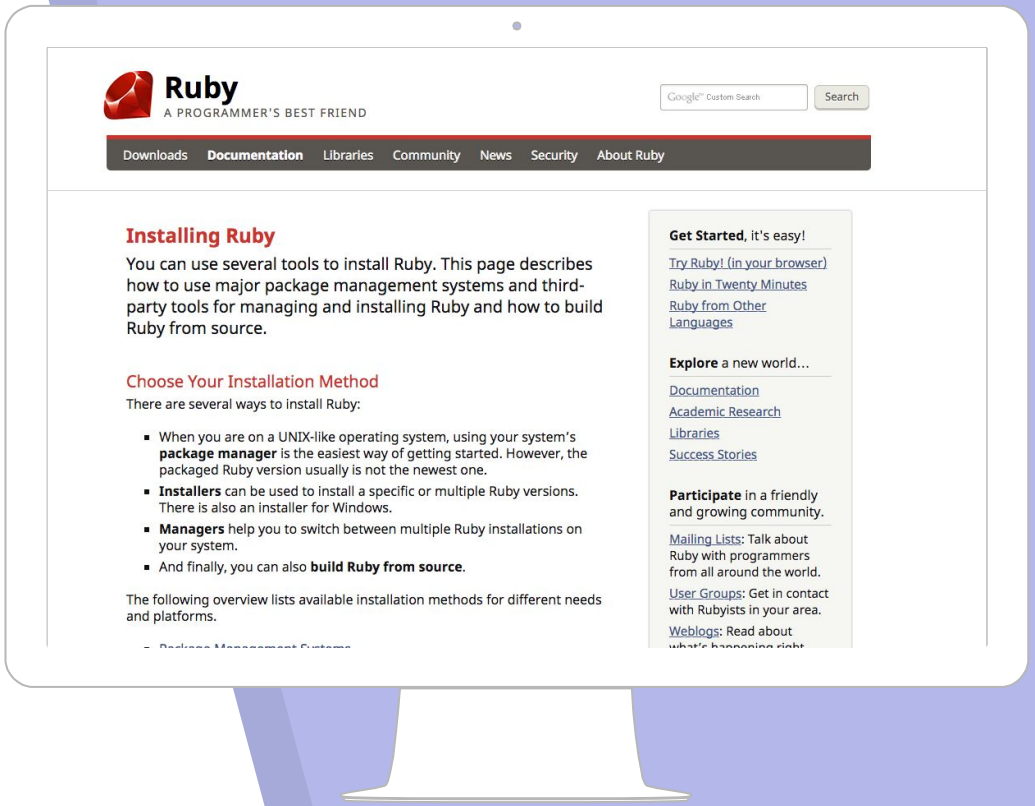
# Instalación de Ruby

# Formas de instalar:

**ruby-lang.org**

- **Package Manager** (versión antigua)
- Rbenv & rvm
- Instaladores
- Compilar las fuentes

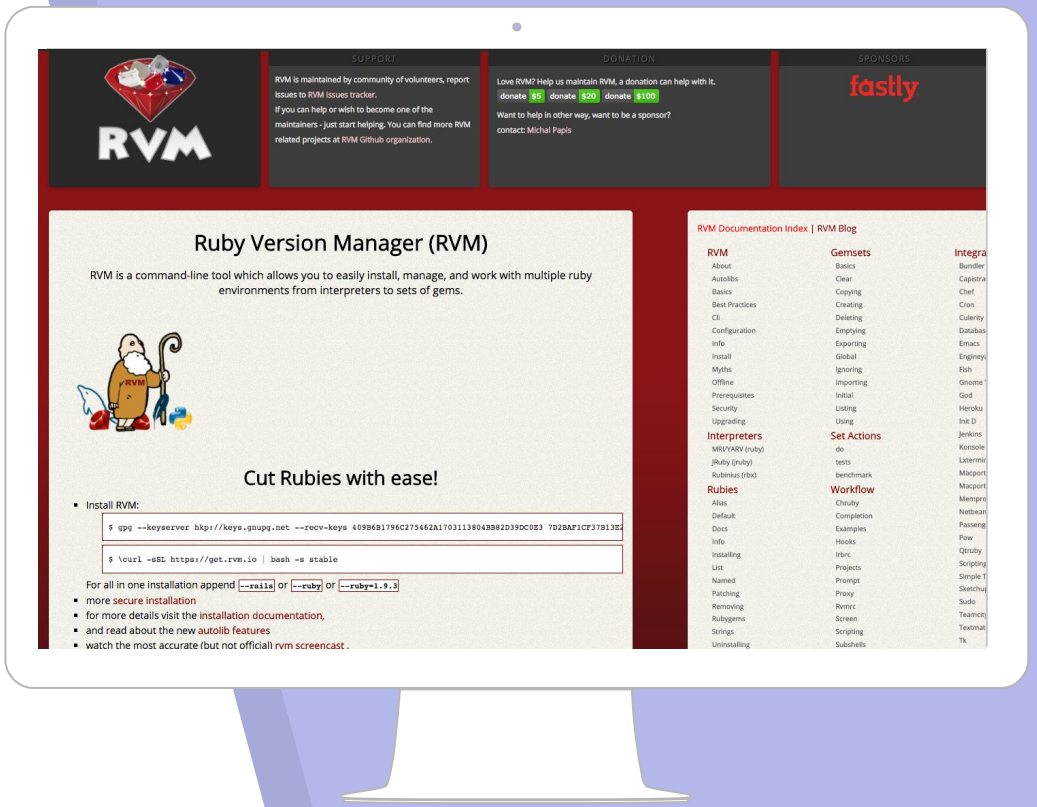
www.ruby-lang.org



# Usando RVM

- Ruby Version Manager
- Instalar **multiples versiones**
- Organizar gemas en **gemsets**

# www.rvm.io

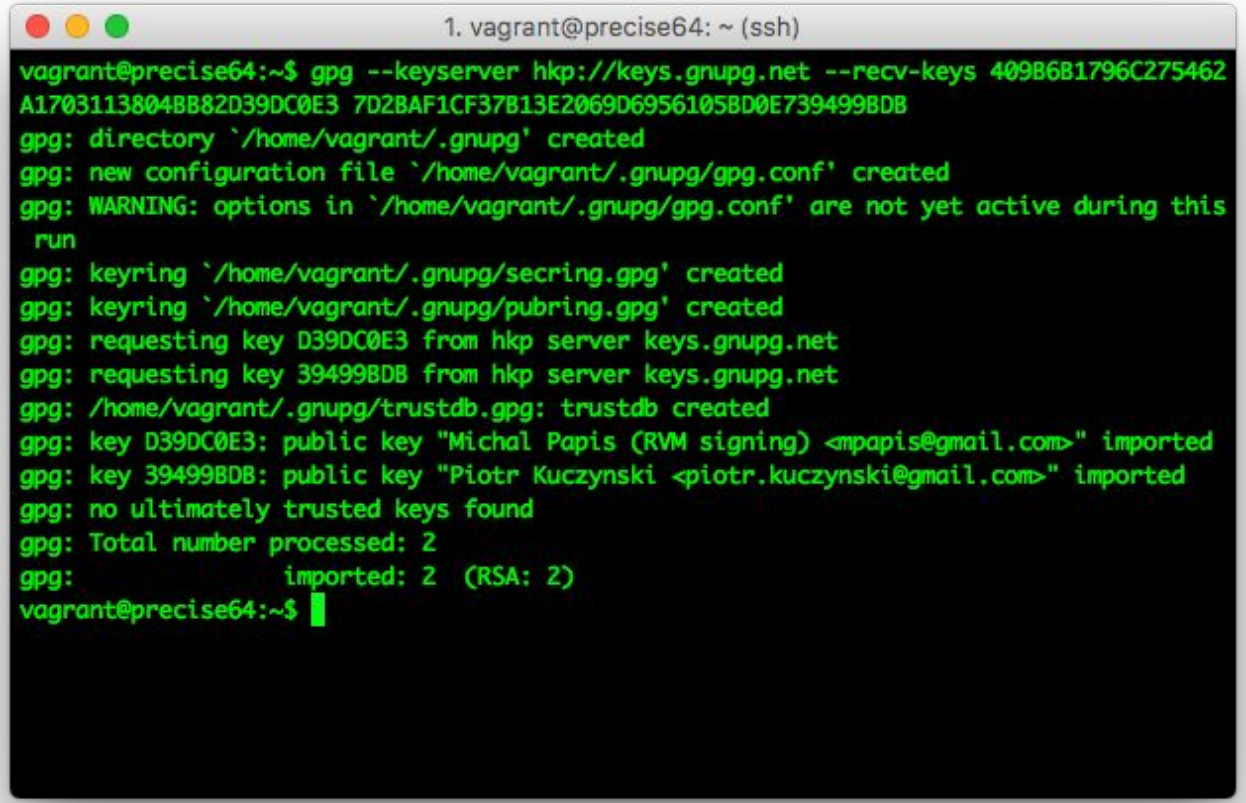

RVM
SUPPORT
RVM is maintained by community of volunteers, report issues to RVM issues tracker.
If you can help or wish to become one of the maintainers - just start helping. You can find more RVM related projects at RVM Github organization.
DONATION
Love RVM? Help us maintain RVM, a donation can help with it.
donate $5 donate $20 donate $100
Want to help in other way, want to be a sponsor?
contact: Michal Papis
SPONSORS
fastly
Ruby Version Manager (RVM)
RVM is a command-line tool which allows you to easily install, manage, and work with multiple ruby environments from interpreters to sets of gems.
Cut Rubies with ease!
Install RVM:
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2
$ \curl -sSL https://get.rvm.io | bash -s stable
For all in one installation append --rails or --ruby or --ruby=1.9.3
more secure installation
for more details visit the installation documentation,
and read about the new autolib features
watch the most accurate (but not official) rvm screencast .
RVM Documentation Index | RVM Blog
RVM
About
Autolibs
Basics
Best Practices
CI
Configuration
Info
Install
Myths
Offline
Prerequisites
Security
Upgrading
Interpreters
MRI/YARV (ruby)
JRuby (jruby)
Rubinius (rbx)
Rubies
Alias
Default
Docs
Info
Installing
List
Named
Patching
Removing
Rubygems
Strings
Uninstalling
Gemsets
Basics
Clear
Copying
Creating
Deleting
Emptying
Exporting
Global
Ignoring
Importing
Initial
Listing
Using
Set Actions
do
tests
benchmark
Workflow
Chruby
Completion
Examples
Hooks
Irbrc
Projects
Prompt
Proxy
Rvmrc
Screen
Scripting
Subshells
Integra
Bundler
Capistra
Chef
Cron
Culerity
Databas
Emacs
Fish
God
Heroku
Init D
Jenkins
Konsole
Macport
Mempro
Netbean
Passeng
Pow
Qtruby
Scripting
Simple T
Sudo
Teamcity
Textmat
Tk

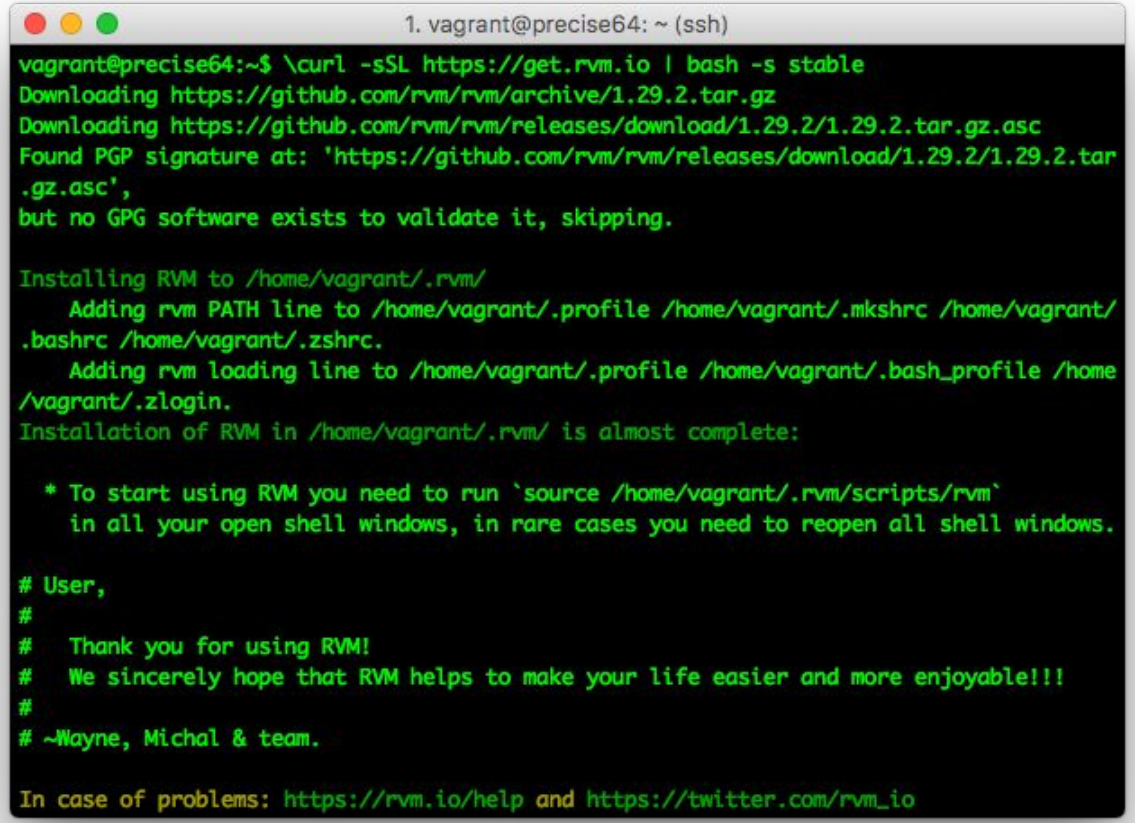
- Install RVM:

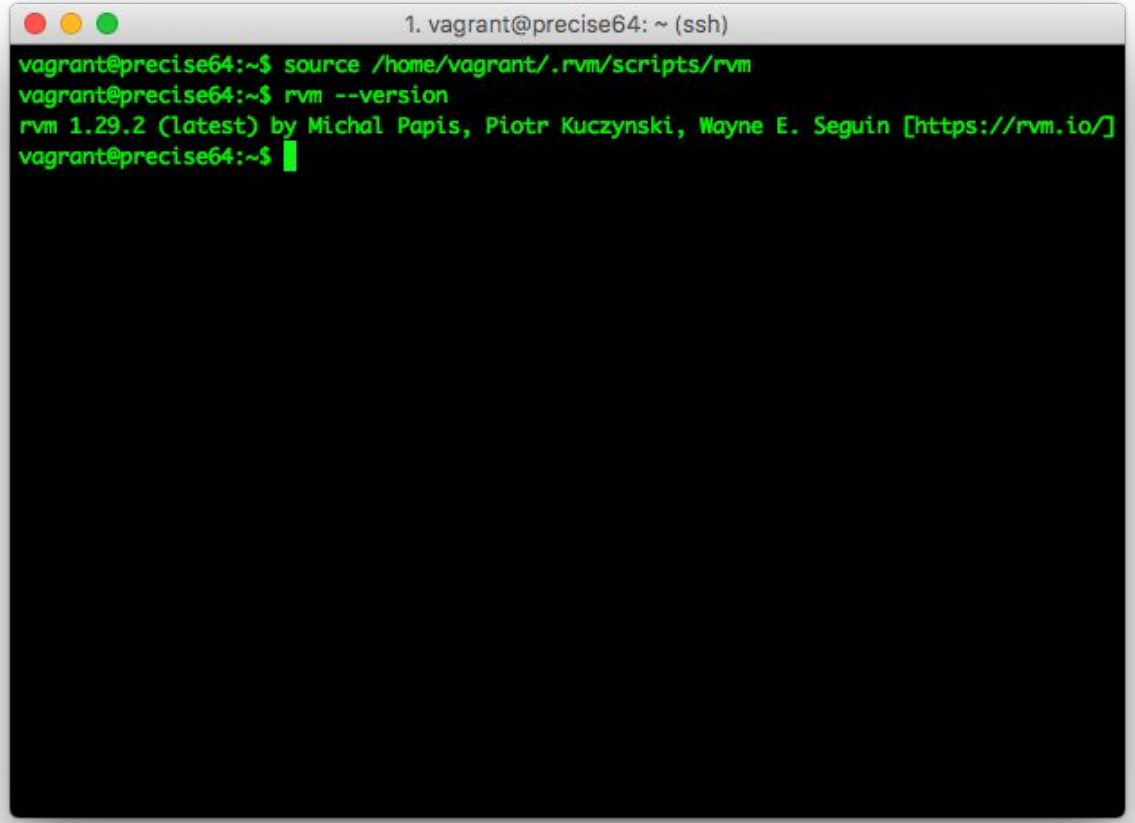
```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2
```

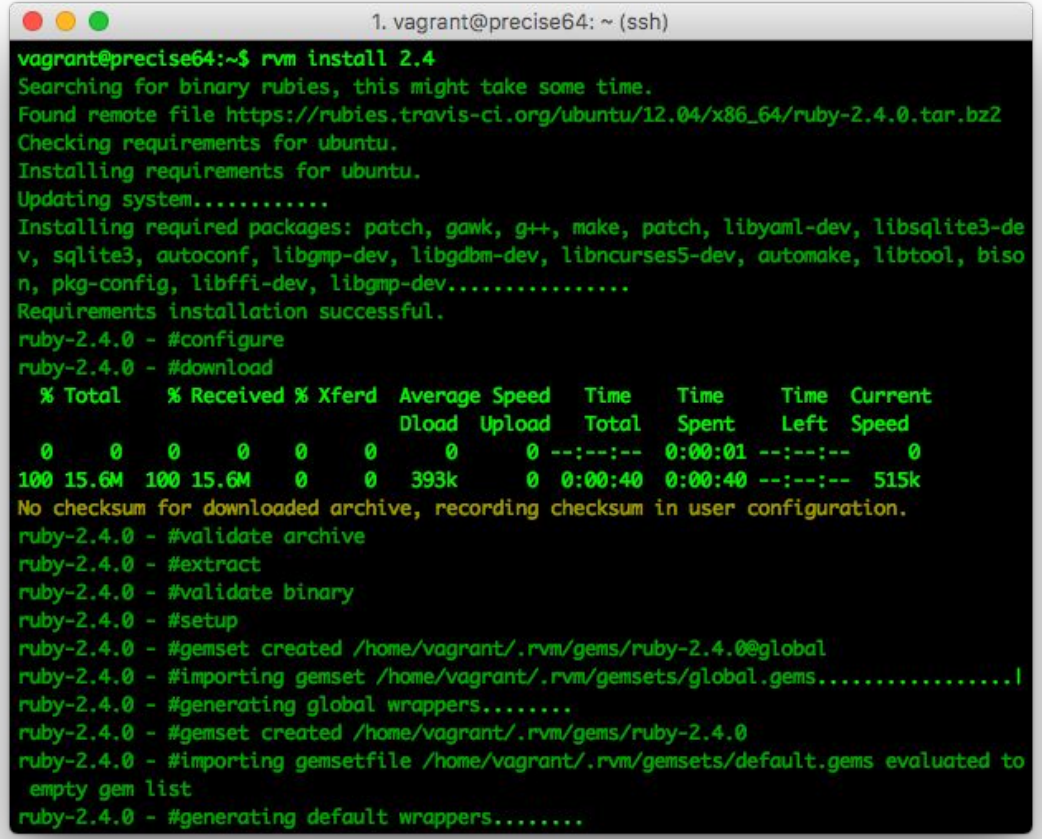
```
$ \curl -sSL https://get.rvm.io | bash -s stable
```

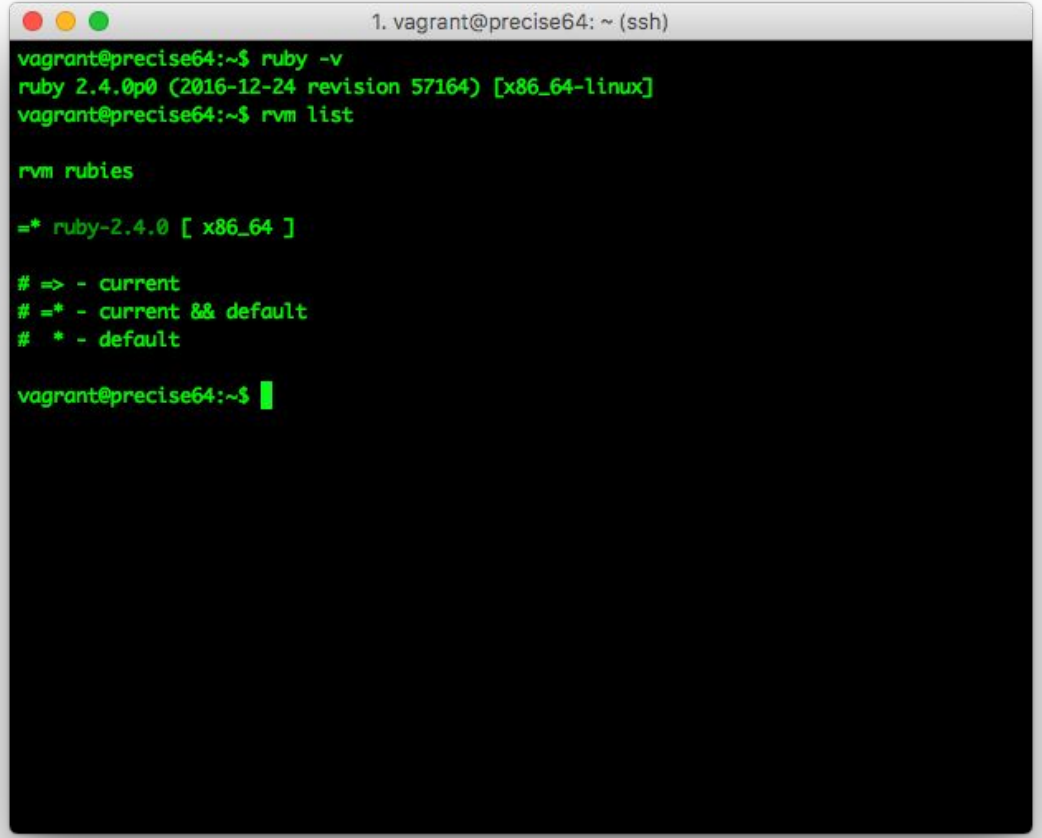
  For all in one installation append `--rails` or `--ruby` or `--ruby=1.9.3`
- more secure installation
- for more details visit the installation documentation,
- and read about the new autolib features
- watch the most accurate (but not official) rvm screencast ,
- read the most accurate (but not official) rvm cheat sheet ,
- starting with Rails? watch the RailsCasts.com on Getting Started with Rails


1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB
gpg: directory `/home/vagrant/.gnupg' created
gpg: new configuration file `/home/vagrant/.gnupg/gpg.conf' created
gpg: WARNING: options in `/home/vagrant/.gnupg/gpg.conf' are not yet active during this run
gpg: keyring `/home/vagrant/.gnupg/secring.gpg' created
gpg: keyring `/home/vagrant/.gnupg/pubring.gpg' created
gpg: requesting key D39DC0E3 from hkp server keys.gnupg.net
gpg: requesting key 39499BDB from hkp server keys.gnupg.net
gpg: /home/vagrant/.gnupg/trustdb.gpg: trustdb created
gpg: key D39DC0E3: public key "Michal Papis (RVM signing) <mpapis@gmail.com>" imported
gpg: key 39499BDB: public key "Piotr Kuczynski <piotr.kuczynski@gmail.com>" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 2
gpg: imported: 2 (RSA: 2)
vagrant@precise64:~$

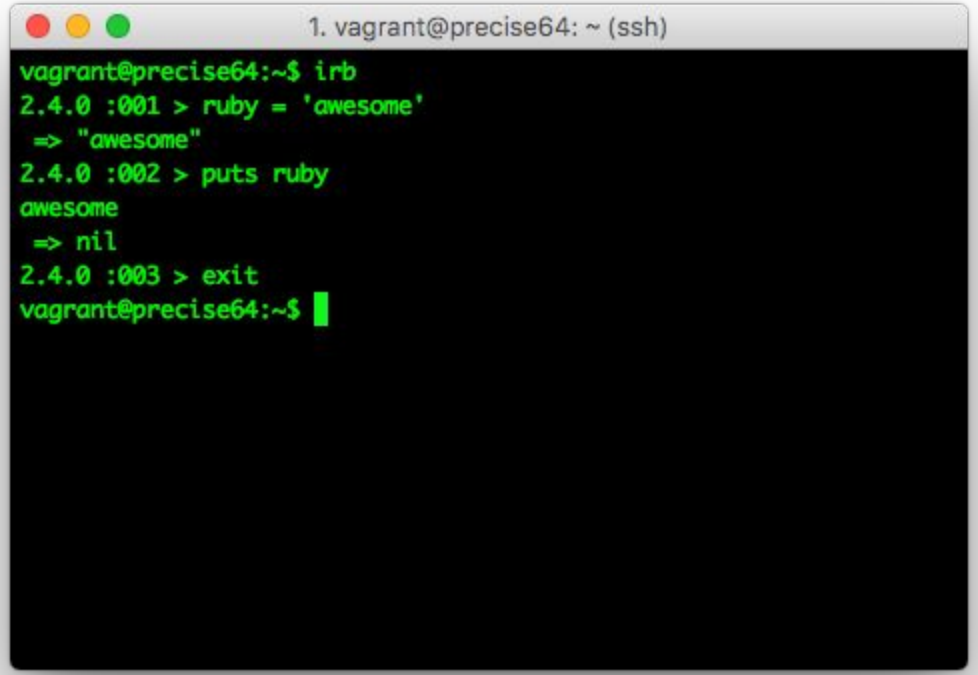

1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ \curl -sSL https://get.rvm.io | bash -s stable
Downloading https://github.com/rvm/rvm/archive/1.29.2.tar.gz
Downloading https://github.com/rvm/rvm/releases/download/1.29.2/1.29.2.tar.gz.asc
Found PGP signature at: 'https://github.com/rvm/rvm/releases/download/1.29.2/1.29.2.tar
.gz.asc',
but no GPG software exists to validate it, skipping.
Installing RVM to /home/vagrant/.rvm/
Adding rvm PATH line to /home/vagrant/.profile /home/vagrant/.mkshrc /home/vagrant/
.bashrc /home/vagrant/.zshrc.
Adding rvm loading line to /home/vagrant/.profile /home/vagrant/.bash_profile /home
/vagrant/.zlogin.
Installation of RVM in /home/vagrant/.rvm/ is almost complete:
* To start using RVM you need to run `source /home/vagrant/.rvm/scripts/rvm`
in all your open shell windows, in rare cases you need to reopen all shell windows.
# User,
#
# Thank you for using RVM!
# We sincerely hope that RVM helps to make your life easier and more enjoyable!!!
#
# ~Wayne, Michal & team.
In case of problems: https://rvm.io/help and https://twitter.com/rvm_io


1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ source /home/vagrant/.rvm/scripts/rvm
vagrant@precise64:~$ rvm --version
rvm 1.29.2 (latest) by Michal Papis, Piotr Kuczynski, Wayne E. Seguin [https://rvm.io/]
vagrant@precise64:~$


1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ rvm install 2.4
Searching for binary rubies, this might take some time.
Found remote file https://rubies.travis-ci.org/ubuntu/12.04/x86_64/ruby-2.4.0.tar.bz2
Checking requirements for ubuntu.
Installing requirements for ubuntu.
Updating system.....
Installing required packages: patch, gawk, g++, make, patch, libyaml-dev, libsqlite3-dev, sqlite3, autoconf, libgmp-dev, libgdbm-dev, libncurses5-dev, automake, libtool, bison, pkg-config, libffi-dev, libgmp-dev.....
Requirements installation successful.
ruby-2.4.0 - #configure
ruby-2.4.0 - #download
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- 0:00:01 --:--:-- 0
100 15.6M 100 15.6M 0 0 393k 0 0:00:40 0:00:40 --:--:-- 515k
No checksum for downloaded archive, recording checksum in user configuration.
ruby-2.4.0 - #validate archive
ruby-2.4.0 - #extract
ruby-2.4.0 - #validate binary
ruby-2.4.0 - #setup
ruby-2.4.0 - #gemset created /home/vagrant/.rvm/gems/ruby-2.4.0@global
ruby-2.4.0 - #importing gemset /home/vagrant/.rvm/gemsets/global.gems.....
ruby-2.4.0 - #generating global wrappers.....
ruby-2.4.0 - #gemset created /home/vagrant/.rvm/gems/ruby-2.4.0
ruby-2.4.0 - #importing gemsetfile /home/vagrant/.rvm/gemsets/default.gems evaluated to empty gem list
ruby-2.4.0 - #generating default wrappers.....


1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ ruby -v
ruby 2.4.0p0 (2016-12-24 revision 57164) [x86_64-linux]
vagrant@precise64:~$ rvm list
rvm rubies
=* ruby-2.4.0 [ x86_64 ]
# => - current
# =* - current && default
#  * - default
vagrant@precise64:~$

# Ruby
# shell!

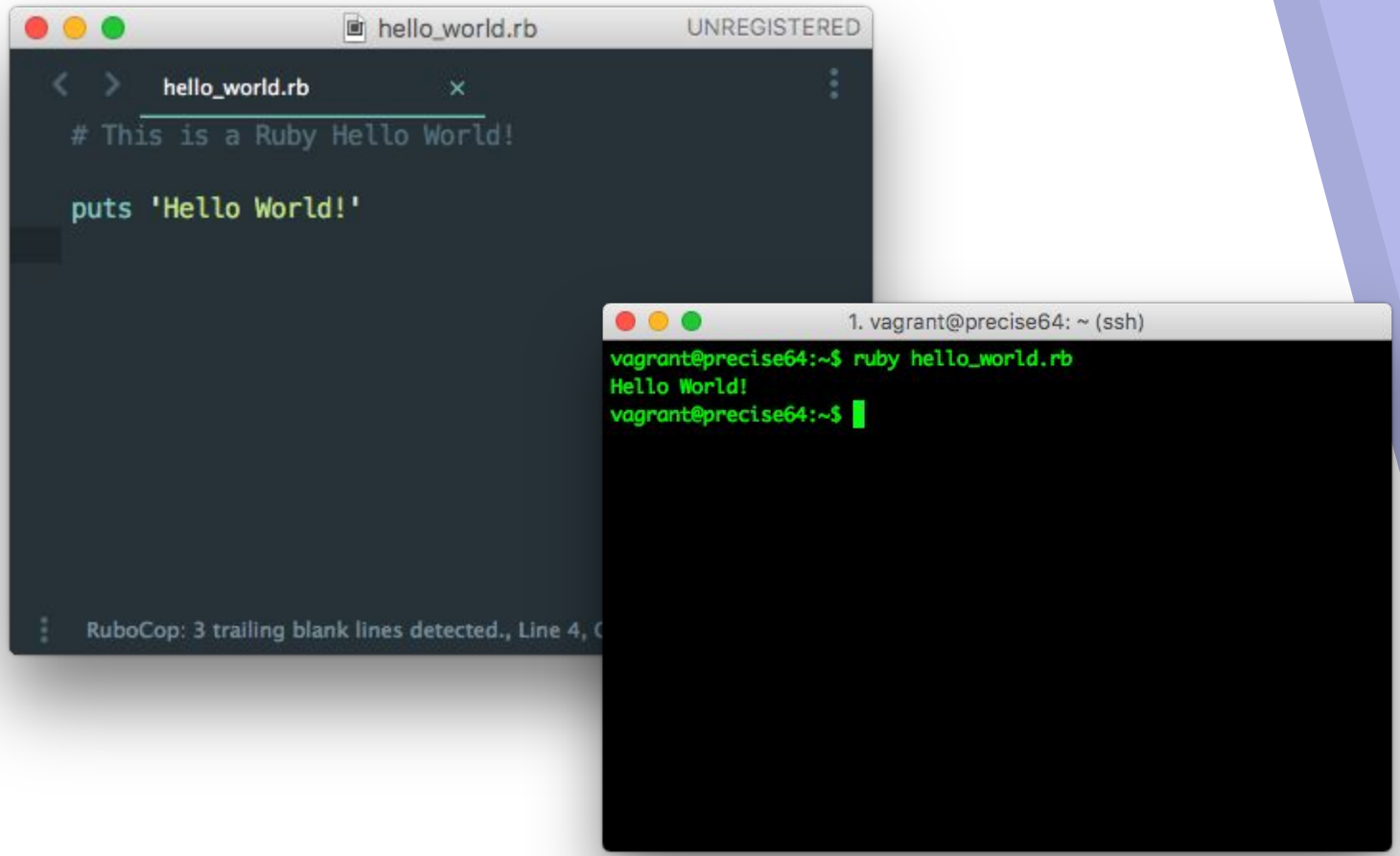

1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ irb
2.4.0 :001 > ruby = 'awesome'
 => "awesome"
2.4.0 :002 > puts ruby
awesome
 => nil
2.4.0 :003 > exit
vagrant@precise64:~$

# Ruby

# ¡Hola Mundo!

# ¿Editor?

- vim
- SublimeText
- Atom
- VS Code
- RubyMine


hello_world.rb
UNREGISTERED
hello_world.rb
# This is a Ruby Hello World!
puts 'Hello World!'
RuboCop: 3 trailing blank lines detected., Line 4,
1. vagrant@precise64: ~ (ssh)
vagrant@precise64:~$ ruby hello_world.rb
Hello World!
vagrant@precise64:~$

# Programación Orientada a Objetos

# Programación Orientada a Objetos (POO)

- **Clases:** Categorías de objetos, definición de una entidad
- **Objetos:** Instancias de las clases con atributos diferentes
  - **Atributos**
  - **Métodos**

# Principios de la Programación Orientada a Objetos.

- Abstracción
- Encapsulación
- Herencia
- Polimorfismo

# Sintaxis de Ruby

github.com/bbatsov/ruby-style-guide

# Sintaxis de Ruby

- Indentamos con **2 espacios**
- suma 3, 4
- metodo_1 metodo_2(param)
- Clases y Módulos **CamelCase**
- Resto **snake_case**

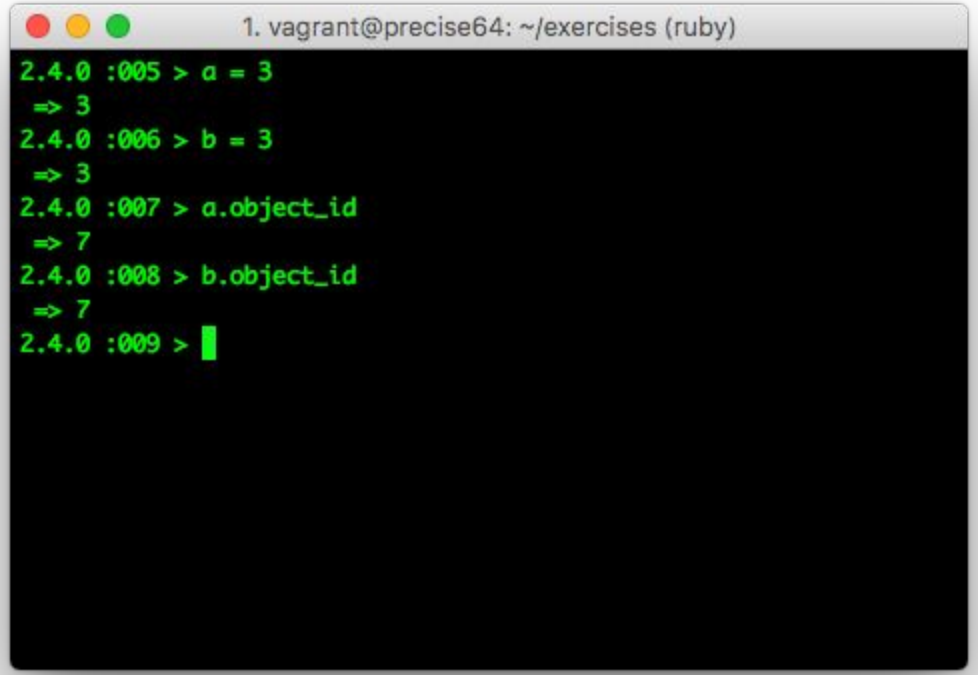
# Variables en Ruby

# Sintaxis

- x
- _x
- name
- my_variable
- myVariable
- var3
- _

# Dinámico

```
my_variable = 'Hello'
# my_variable value is the string 'Hello'

my_variable = 3
# Now the value is the numeric value 3!
```


1. vagrant@precise64: ~/exercises (ruby)
2.4.0 :005 > a = 3
=> 3
2.4.0 :006 > b = 3
=> 3
2.4.0 :007 > a.object_id
=> 7
2.4.0 :008 > b.object_id
=> 7
2.4.0 :009 >

# Ámbito de variables locales

```ruby
def say_hello
  x = 'Hello'
  puts x

  say_goodbye
end

def say_goodbye
  x = 'GoodBye'
  puts x
end

x = 'Let\'s check variable scope'
puts x

say_hello

# Output:
#
# Let's check variable scope
# Hello
# GoodBye
```

# Ámbito de variables locales

```ruby
x = 'hello'

def foo
  puts x
end

foo

# undefined local variable or method `x' for
# main:Object (NameError)
```

# Referencias

```
str = 'Hello'

abc = str

str.replace('Goodbye')

puts str
puts abc

# Goodbye
# Goodbye
```

# Más variables!

- @instance_variables
- $global_variables
- @@class_variables

# Constantes en Ruby

OPENWEBINARS = 10

MY_CONSANT = ‘hello’

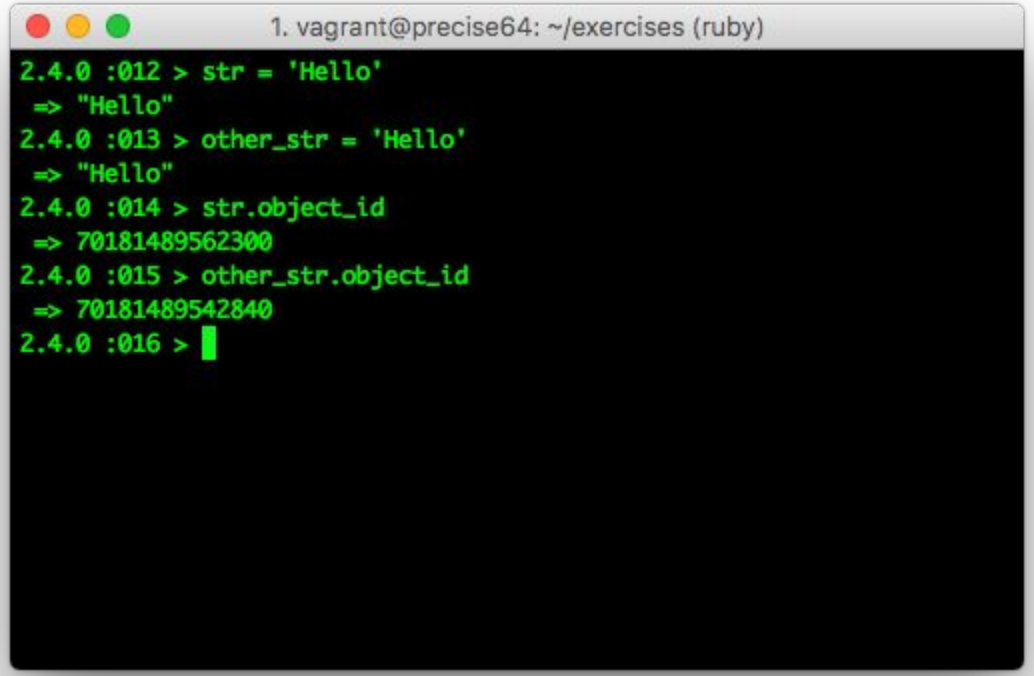
# Strings en Ruby

# Strings

Los strings son **objetos**

- “Esto es un string.”
- ‘Esto también es un string’

- Interpolación:
  - “Hola #{name}”
  - “La solución es #{2 + 2}”


1. vagrant@precise64: ~/exercises (ruby)
2.4.0 :012 > str = 'Hello'
=> "Hello"
2.4.0 :013 > other_str = 'Hello'
=> "Hello"
2.4.0 :014 > str.object_id
=> 70181489562300
2.4.0 :015 > other_str.object_id
=> 70181489542840
2.4.0 :016 >

# Manipular Strings

```
>> str = ‘Ruby es fantástico!’
>> str.upcase
=> “RUBY ES FANTÁSTICO!”
>> str.reverse
=> “!ocitsátnaf se ybuR”
```

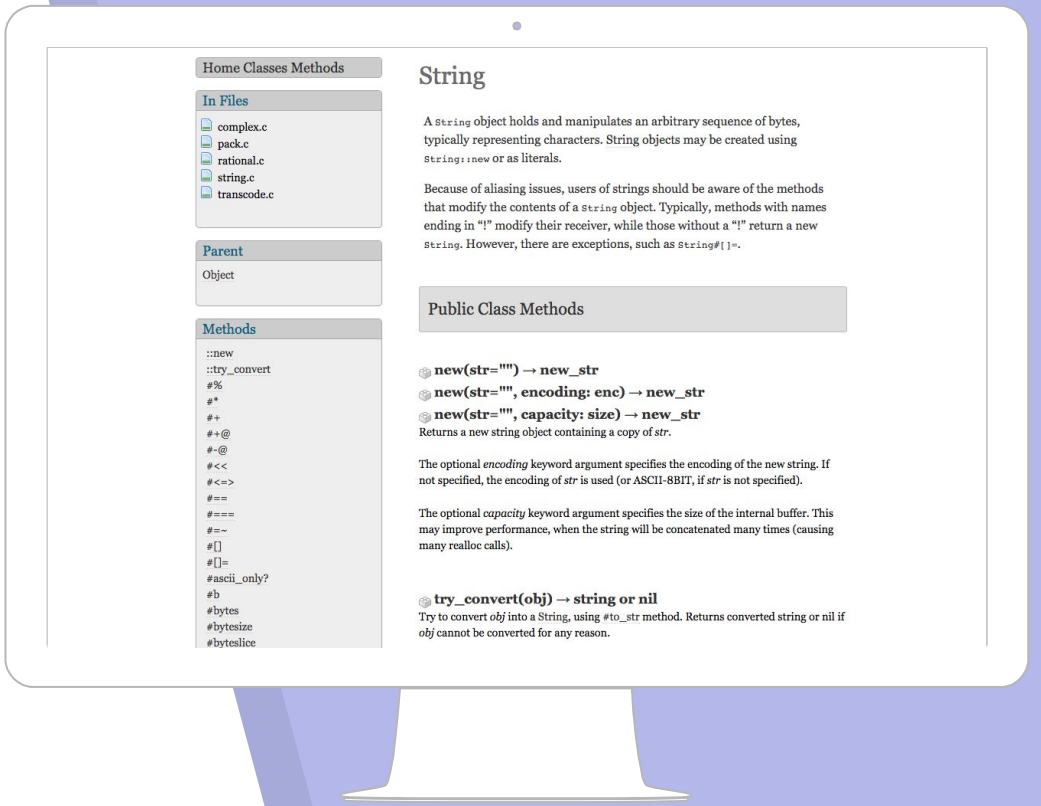
# Strings - Métodos de consulta

```
>> str = ‘Ruby is awesome!’
>> str.include?(‘Ruby’)
=> true
>> str.include?(‘Java’)
=> false
>> str.start_with?(‘Ruby’)
=> true
>> str.size
=> 15

>> “”.empty?
=> true
```

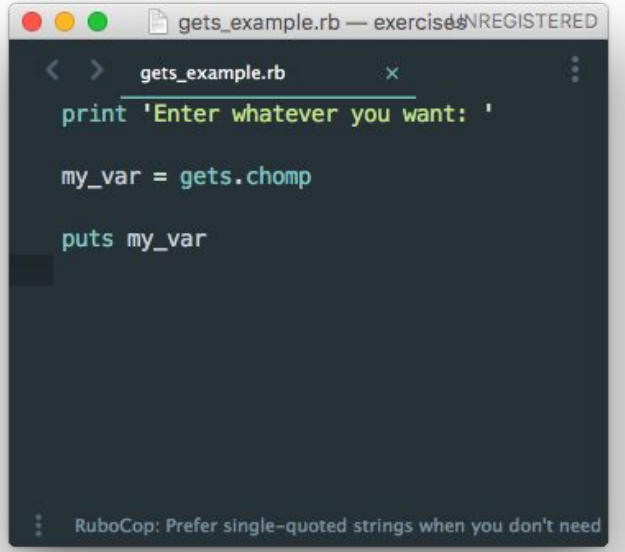
# Comparar strings

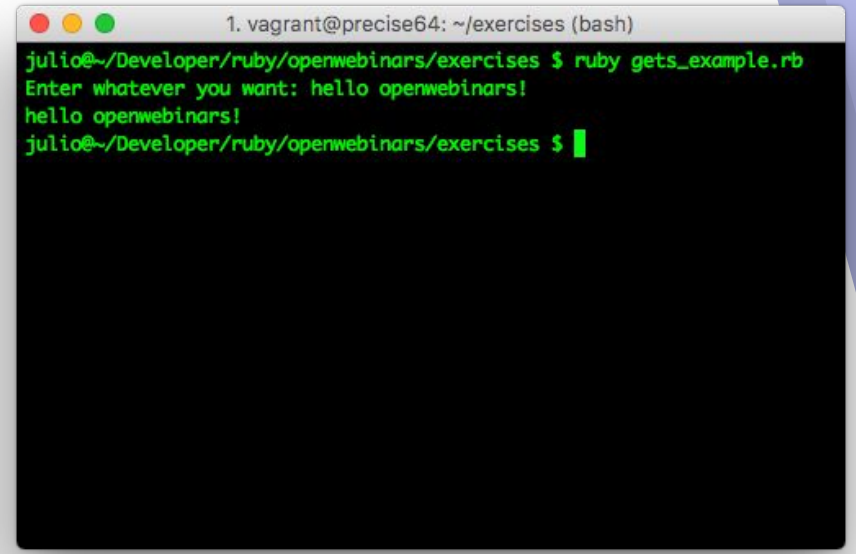
```
>> “Hello” == “Hello"   # ‘Hello’.==(‘Hello’)
=> true
>> “Hello” == “HELLO”
=> false
>> “Hello”.equal?(“Hello”)
=> false

>> a = “Hello”
>> b = a
>> a.equal?(b)
=> true
```

www.ruby-doc.org



# ¡Conoce a gets!



# Ruby
# Symbols

# Symbols

```
>> my_symbol = :hello
=> :hello
>> my_symbol = :”symbol with spaces!”
=> :”symbol with spaces!”
```

# Symbols son únicos

```
>> str_a = “hello”
>> str_b = “hello”
>> str_a.object_id == str_b.object_id
=> false

>> symbol_a = :hello
>> symbol_b = :hello
>> symbol_a.object_id == symbol_b.object_id
=> true
```

# Números en Ruby

# Objetos numéricos

```
>> num = 99
>> num = 99.6
>> num.round
=> 100
>> num.zero?
=> false
```

# Objetos numéricos

- Numeric
  - Float
  - Integer
    - Fixnum
    - Bignum

# Operaciones aritméticas

```
>> 1 + 1 # 1.+(1)
=> 2
>> 10 / 5 # 10./(5)
=> 2
>> 16 / 5
=> 3
>> 16.0 / 5
=> 3.2
```

# Operadores en Ruby

# Operadores aritméticos

```
>> a + b
>> a.+(b)

>> a + b
>> a - b
>> a * b
>> a / b
>> a % b
>> a ** b
```

# Operadores (métodos) de comparación

```
>> a == b # a.==(b)
>> a != b # a.!=(b)
>> a > b
>> a < b
>> a >= b
```

# Operadores de asignación

```
>> a = 1
>> a += 1 # a = a + 1
=> 2
>> -=
>> *=
>> /=
>> %=
>> **=

>> a, b, c = 10, 20, 30
>> a, b, c = [10, 20, 30]
```

# Operadores lógicos

```
>> &&
>> ||
>> and
>> or
>> !
>> not
```

# Operadores lógicos

```
>> a = true and false
=> false
>> a
=> true
>> a = true && false
=> false
>> a
=> false
```

# Operador ternario

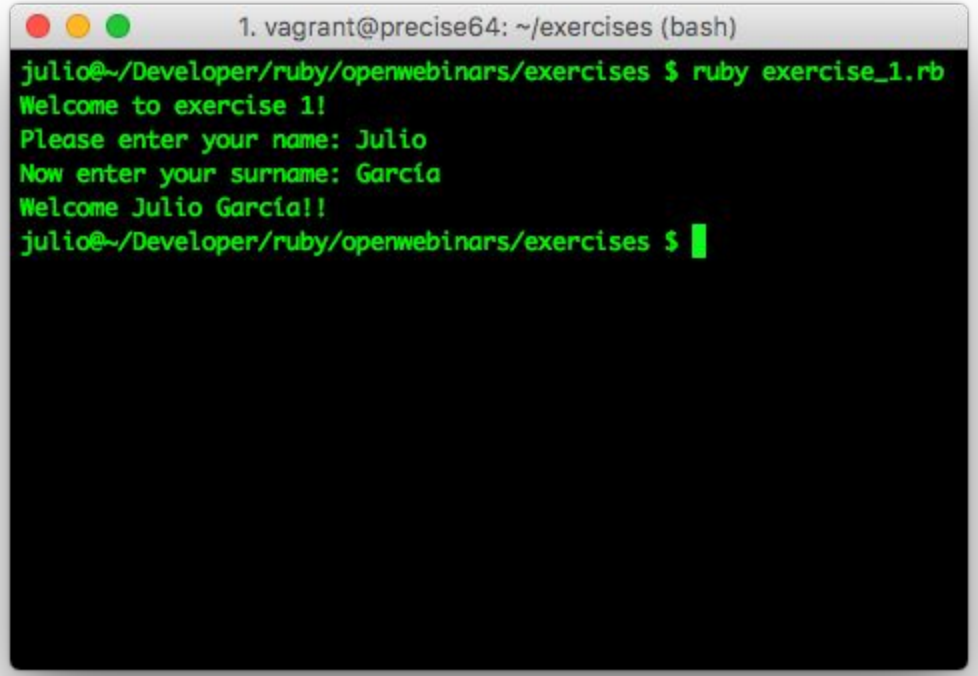
```
>> true == false ? 1 : 2
=> 2

# Example:
value = age > 18 ? 20 : 10
```

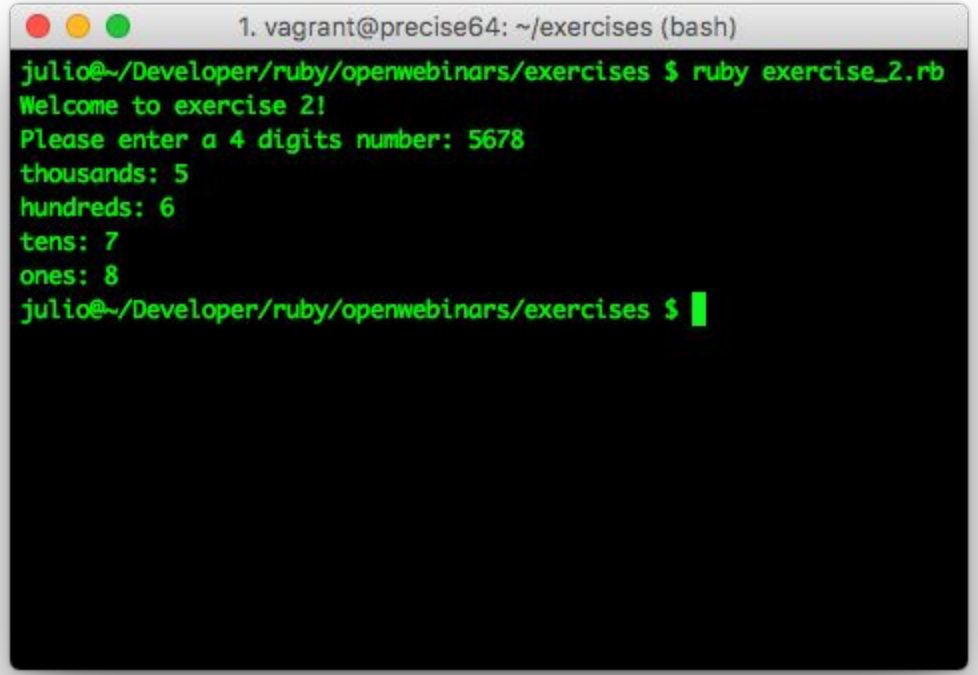
# ¡Resumen!

- **Todo** es un objeto
- Sintaxis
- Variables
- Strings
- Symbols
- Números
- Operadores

¡Vamos a practicar!

# Let’s practice!


1. vagrant@precise64: ~/exercises (bash)
julio@~/Developer/ruby/openwebinars/exercises $ ruby exercise_1.rb
Welcome to exercise 1!
Please enter your name: Julio
Now enter your surname: García
Welcome Julio García!!
julio@~/Developer/ruby/openwebinars/exercises $


1. vagrant@precise64: ~/exercises (bash)
julio@~/Developer/ruby/openwebinars/exercises $ ruby exercise_2.rb
Welcome to exercise 2!
Please enter a 4 digits number: 5678
thousands: 5
hundreds: 6
tens: 7
ones: 8
julio@~/Developer/ruby/openwebinars/exercises $

# Colecciones y Objetos contenedores

# Array vs Hash

```
# Array
>> [1, ‘element’, :hey]

# Hash
>> { my_key: 1, the_element: ‘element’, the_key: :hey }
```

# Arrays en Ruby

# Crear un nuevo array

```
>> Array.new
=> []
>> Array.new(3)
=> [nil, nil, nil]
>> Array.new(3, ‘hey!’)
=> [‘hey!’, ‘hey!’, ‘hey!’]

>> a = []
>> a = [1, 2, ‘three’, :four, []]
```

# Insertar y recibir ratos de array

```
>> a = []
>> a.[]=(0, ‘first’)
>> a[0] = “first”
>> a = [1, 2, 3, 4]
>> a.[](2)
=> 3
>> a[2]
=> 3
>> a[9] = 10
>> a
=> [1, 2, 3, 4, nil, nil, nil, nil, nil, 10]
```

# Insertar y recibir ratos de array

```
[]← push     []→ pop      ←[] shift        →[] unshift

>> a = [1,2,3,4]
>> a.push(5)
=> [1,2,3,4,5]
>> a.pop           # a is [1,2,3,4]

>> a << 5
=> [1, 2, 3, 4, 5]
```

# Combinar arrays

```
>> a = [1, 2, 3]
>> a.concat([4, 5, 6])          # a is modified!
>> b = a.+([4, 5, 6])
>> b = a + [4, 5 ,6]            # b is a new array

>> a = [1, 2, 3]
>> b = a                        # a.object_id == b.object_id
>> a.replace([4, 5, 6])
>> a
=> [4, 5, 6]
>> b
=> [4, 5, 6]
```

# Transformar arrays

```
>> a = [1, 2, [3, 4, [5], [6, [7, 8]]]]
>> a.flatten
=> [1, 2, 3, 4, 5, 6, 7, 8]
>> a
=> [1, 2, [3, 4, [5], [6, [7, 8]]]]
```

# Transformar arrays

```
a = [1, 2, [3, 4, [5], [6, [7, 8]]]]

# a becomes a new object
a = a.flatten

# a is the same object
a.flatten!
```

# Transformar arrays

```
>> a = [1, 2, [3, 4, [5], [6, [7, 8]]]]
>> a.object_id
=> 70097305707820
>> a.flatten!
=> [1, 2, 3, 4, 5, 6, 7, 8]
>> a.object_id
=> 70097305707820
```

# Transformar arrays

```
>> a = [1, 2, 3, 4, 5]
>> a.reverse
=> [5, 4, 3, 2, 1]
>> [2,4,1].sort
=> [1, 2, 4]
>> a.join('-')
=> '1-2-3-4-5'

>> a = [1, 2, 3, 1, 4, 5, 1]
>> a.uniq
=> [1, 2, 3, 4, 5]
```

# Transformar arrays

```
# ¡Son referencias a objetos!

>> a = [1, 2, 3, 4, 5]
>> b = a
>> b[2] = ‘changed!’
>> a
=> [1, 2, ‘changed!’, 4, 5]
```

# Transformar arrays

```
# El uso de .dup

>> a = [1, 2, 3, 4, 5]
>> b = a.dup
>> a[2] = ‘changed!’
>> b
=> [1, 2, 3, 4, 5]
```

# Array - Métodos de consulta

```
a.size          # a.length
a.empty?
a.include? item
a.count         # a.count(1)
a.first         # a.first(4)
a.last
```

# Hashes en Ruby

# Ejemplo Ruby Hash

```
countries = { ‘Spain’ => ‘ES’, ‘France’ => ‘FR’ }

puts ‘Enter the name of a country’
country = gets.chomp
code = countries[country]
puts “The country code for #{country} is #{code}”
```

# Nuevo hash

```
>> a = Hash.new(‘default_value’)
=> {}
>> a[:hello]
=> ‘default_value’

>> a = {}
>> a.[]=(:hello, ‘Hello friends!’)
>> a[:hello] = ‘Hello friends!’
>> a
=> { :hello => ‘Hello friends!’ }
```

# Obtener un valor de un hash

```
>> a = { a: 1, b: 2 }
>> a[:c] = 3
>> a[:b]
=> 2
>> a.[](:c)
=> 3
>> a[:d]
=> nil
```

# Combinar hashes

```
>> a = { ‘Smith’ => ‘John’, ‘Jones’ => ‘Jane’ }
>> b = { ‘Smith’ => ‘Jim’ }
>> a.merge! b
>> a[‘Smith’]
=> ‘Jim’
>> c = a.merge { ‘Potter’ => ‘Harry’ }
=> { ‘Smith’ => ‘Jim’, ‘Jones’ => ‘Jane’, ‘Potter’ => ‘Harry’ }
```

# Transformaciones en hashes

```
>> a = { a: ‘1’, b: ‘2’, c: ‘3’ }
>> a.invert
=> { ‘1’ => :a, ‘2’ => :b, ‘3’ => :c }

>> a[:a] = ‘11’
>> a
=> { a: ‘11’, b: ‘2’, c: ‘3’ }
```

# Hash - Métodos de consulta

```
a.has_key?(:a)
a.empty?
a.size
a.keys
a.values
```

# Built-in

## to_*

# to_* - métodos de conversión de tipo

```
>> 3.to_s
=> ‘3’
>> “#{3}”
>> ‘100’.to_i
=> 100
>> 100.to_f
=> 100.0
>> 100.3.to_i
=> 100
```

# to_* - métodos de conversión de tipo

```
>> [1, 2, 3].to_s
=> “[1, 2, 3]”
>> [[:a, 1], [:b, 2]].to_h
=> {:a=>1, :b=>2}

>> { a: 1, b: 2 }.to_a
=> [[:a, 1], [:b, 2]]
```

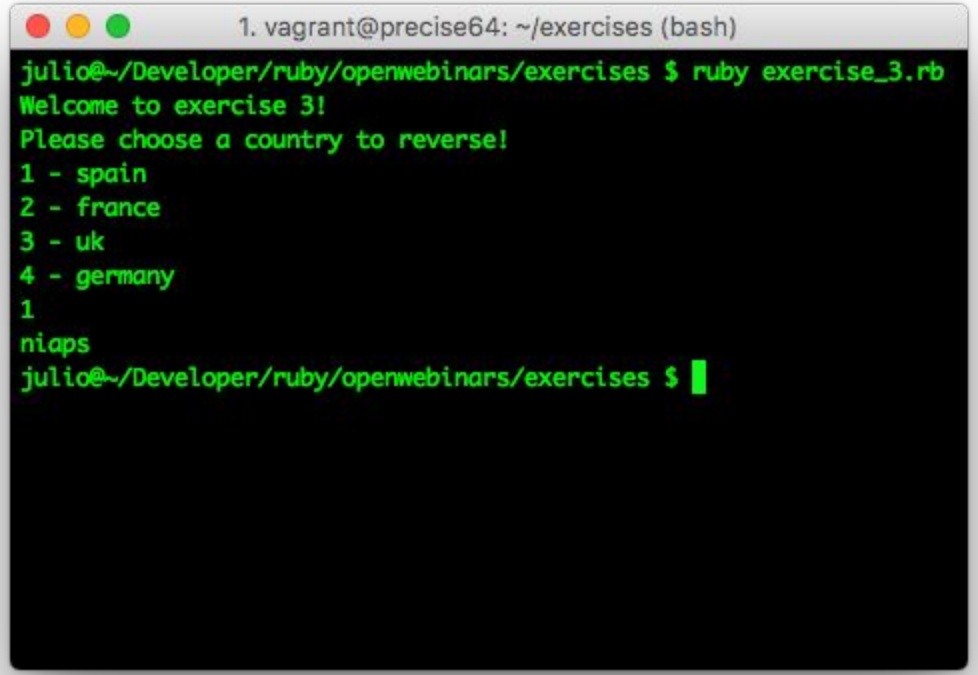
# ¡Resumen!

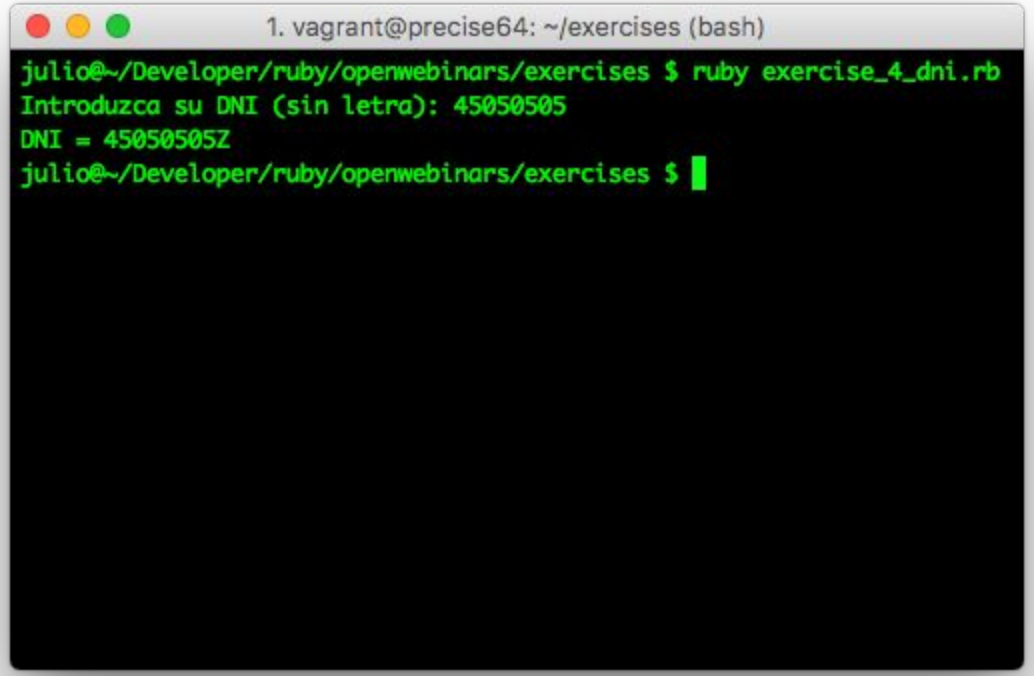
- Objetos contenedores Built-In
  - Hash
  - Array
- Métodos de transformación
- Métodos de consulta

- to_* conversiones de tipo

¡Vamos a practicar!

# Let’s practice!


1. vagrant@precise64: ~/exercises (bash)
julio@~/Developer/ruby/openwebinars/exercises $ ruby exercise_3.rb
Welcome to exercise 3!
Please choose a country to reverse!
1 - spain
2 - france
3 - uk
4 - germany
1
niaps
julio@~/Developer/ruby/openwebinars/exercises $




| RESTO | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LETRA | T | R | W | A | G | M | Y | F | P | D | X | B |

| RESTO | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LETRA | N | J | Z | S | Q | V | H | L | C | K | E |

NUMERO / 23

# Ruby
# Blocks

# Ruby blocks

```
def block_test
  puts ‘the method’
  yield if block_given?
  puts ‘the method again’
end

block_test do
  puts ‘the block!’
end

# block_test { puts ‘the block! }
```

# Ruby blocks

```
def block_test(&block)
  puts ‘the method’
  block.call if block_given?
  puts ‘the method again’
end

block_test do
  puts ‘the block!’
end

# block_test { puts ‘the block! }
```

# Ruby blocks

```
the method
the block!
The method again
```

# Ruby blocks

```
def block_test
  yield(1)
  yield(2)
  yield(3)
end

block_test do |num|
  puts “num is #{num}”
end

block_test { |num| }
```

# Ruby blocks

```
num is 1
num is 2
num is 3
```

# Ruby procs & lambdas

```
a = Proc.new { puts ‘A!’ }
a = proc { puts ‘A!’ }
a.call

b = lambda { puts ‘A!’ }
b = ->{ puts ‘A’ }
b.call
```

# Iteradores en Ruby

# Iteradores en Ruby - **each**

```
a = [1, 2, 3, 4]

a.each do |element|
  puts element
end
```

```
1
2
3
4
```

# Iteradores en Ruby - **each**

```
a = { a: 1, b: 2, c: 3, d: 4}

a.each do |key, value|
  puts “#{key}, #{value}”
end
```

```
a, 1
b, 2
c, 3
d, 4
```

# Iteradores en Ruby - **map**

```
a = [1, 2, 3, 4]

a.map do |element|
  return element - 1
end

=> [0, 1, 2, 3]

>> a.map! { |el| el + 1 }
>> a
=> [0, 1, 2, 3]
```

# Iteradores en Ruby - **map**

```
a = [1, 2, 3, 4]

a.map do |element|
  { element: element }
end

=> [{ element: 1 }, { element: 2 }, ...]
```

# Iteradores en Ruby - **select**

```
a = [1, 2, 3, 4]

a.select do |element|
  element > 2
end

=> [3, 4]
```

# Iteradores en Ruby - **reject**

```
a = [1, 2, 3, 4]

a.reject do |element|
  element > 2
end

=> [1, 2]
```

# Iteradores en Ruby - **each_with_object**

```
a = [1, 2, 3, 4]

a.each_with_object({}) do |element, hash|
  hash[element.to_s] = element
end

>> { "1"=>1, "2"=>2, "3"=>3, "4"=>4 }
```

# Estructuras de control en Ruby

# Estructuras de control de flujo

- Ejecución **condicional**
- Looping
- Métodos
- Excepciones

# Ejecución condicional de código

- Acceder **SI** la contraseña es correcta
- Error **A NO SER** que exista un producto

**Las decisiones** basadas en **condiciones** son tan **comunes** en programación como en la vida misma.

# IF y sus amigos

# IF - ELSE

```
if condition
  # code here
end
```

# IF - ELSE

```
if condition
  # code here
else
  # code here
end
```

# IF - ELSE

```
if age > 18
  sell_beer
elsif age < 9
  free_juice
else
  sell_juice
end
```

# IF vs **UNLESS**

```
if !(age > 18)
  forbidden
end

unless (age > 18)
  forbidden
end
```

# ¡En una línea!

```
forbidden if !(age > 18)
```

```
forbidden unless age > 18
```

# Bloques **case**

```
puts ‘Exit the program? (yes or no):’
answer = gets.chomp

case answer
when ‘yes’
  puts ‘Good-bye!’
  exit
when ‘no’
  puts ‘OK, we\’ll continue’
else
  puts ‘unknown answer’
end
```

# Bloques **case**

```
puts ‘Exit the program? (yes or no):’
answer = gets.chomp

if answer == ‘yes’
  puts ‘Good-bye!’
  exit
elif answer == ‘no’
  puts ‘OK, we\’ll continue’
else
  puts ‘unknown answer’
end
```

# Bloques **case**

```
puts ‘Exit the program? (yes or no):’
answer = gets.chomp

case answer
when ‘y’, ‘yes’
  puts ‘Good-bye!’
  exit
when ‘n’, ‘no’
  puts ‘OK, we\’ll continue’
else
  puts ‘unknown answer’
end
```

# Let’s practice!

# Repetición con bucles

# Loop con loop

```
loop { puts ‘looping forever!’ }

loop do
  puts ‘looping forever’
end
```

# Break con break

```
n = 1
loop do
  n += 1
  break if n > 9
end
```

# next

```
n = 1
loop do
  n += 1
  next unless n == 100
  break
end
```

# Looping condicional

```
n = 1
while n < 11
  puts n
  n += 1
end

n+= 1 while n < 11      # also one-line!
```

# Conditional looping

```
n = 1
until n >= 11
  puts n
  n += 1
end

n+= 1 until n >= 11          # also one-line!
```

# Recuerdas los blocks?

```
def my_loop
  yield while true
end

my_loop { ‘Looping forever!’ }
```

# ¡Repite N veces!

```
10.times do |i|
  puts i
end

arr.length.times do
  ...
end
```

# Let’s practice!

# Métodos en Ruby

# Métodos en Ruby

```
def my_method
  puts “I’m in a method”
end

# some code...
...
my_method
...
```

# Métodos en Ruby

```
def my_sum(num_a, num_b)
  return num_a + num_b
end

result = my_sum(1, 2)
puts “1 + 2 = #{result}”

=> 1 + 2 = 3
```

# Métodos en Ruby

```
def my_sum(num_a, num_b)
  num_a + num_b
End

result = my_sum 1, 2
puts “1 + 2 = #{result}”

=> 1 + 2 = 3
```

# Métodos en Ruby - **argumentos**

```ruby
def my_method(*args)
  args.sum
end

my_method(1, 2, 3)
=> 6
my_method(1, 2)
=> 3
```

# Métodos en Ruby - **argumentos**

```
def value(discount = 0)
  cost = 10 * (1 - discount)

  “#{cost} €”
end

>> value
=> 10 €
>> value 0.2
=> 8.0 €
```

# Métodos en Ruby - **argumentos**

```
def profile(name, surname, address, tel, work, food)
  ...
end

profile(...)
```

# Métodos en Ruby - **argumentos**

```
def profile(name:, surname:, address:, tel:, work:, food:)
  ...
end

profile(name: ‘John’, tel: ‘666’, ...)
```

# Métodos en Ruby - **argumentos**

```
def profile(name: ‘Julio’)
  puts name
end

profile
=> ‘Julio’
Profile ‘John’
‘John’
```

# El argumento **block**

```
def foo(*args, &block)
  ...
End
```

# El argumento **block**

```
def foo
  if block_given?
    …
    yield(val)
    …
  else
    …
  end
end
```

# Anidar métodos

```
>> :my_symbol.to_s.upcase.split(‘_’).push(1).join(‘-’)
=> “MY-SYMBOL-1”

# “my_symbol”
# “MY_SYMBOL”
# [“MY”, “SYMBOL”]
# [“MY”, “SYMBOL”, 1]
# “MY-SYMBOL-1”
```

# Let’s practice!

# Let’s practice - **max**

```
# Nombre del método: max
# Entrada: Lista de números
# Devuelve: El número más grande
# Imprime: Nada
```

# Let’s practice - **longest_string**

```
# Nombre del método: longest_string
# Entrada: Lista de strings
# Devuelve: La cadena más larga
# Imprime: Nada
```

# Let’s practice - **word_count**

```
# Nombre del método: word_count
# Entrada: string con varias palabras
# Devuelve: número de palabras
# Imprime: Nada
```

# Let’s practice - **sum**

```
# Nombre del método: sum
# Entrada: Lista de números
# Devuelve: suma de todos los números
# Imprime: Nada
```

# Let’s practice - **mean**

```
# Nombre del método: sum
# Entrada: Lista de números
# Devuelve: La media aritmética
# Imprime: Nada
```

# Let’s practice - **hot_or_cold**

```
# Nombre del método: hot_or_cold
# Comportamiento: Adivinar un número
# Entrada: número para adivinar
# Imprime: > o <
```

# Let’s practice - **find_even**

```
# Nombre del método: find_even
# Entrada: Lista de números
# Devuelve: Lista con todos los números pares
# Imprime: Nada
```

# Qué es true?

# ¡**Toda** expresión se evalua!

```
>> a = 1
=> 1
>> [1, 2].pop
=> 2
>> i += 1 while i <= 10
=> nil
```

# ¡**Toda** expresión se evalua!

```
>> 2 > 1
=> true
>> 2 < 1
=> false
```

# Estados True o false

```
>> puts ‘hello’ if (def method; end)
hello
>> puts ‘hello’ if ‘string’
hello
>> puts ‘hello’ if 30
Hello
```

| **Expression** | **Object** to which expression evaluates | **Boolean** value of expression |
|---|---|---|
| 1 | 1 | True |
| 0 | 0 | True |
| 1+1 | 2 | True |
| true | true | True |
| false | false | False |
| 100 > 50 | true | True |
| “” | “” | True |
| puts “string” | nil | False |

# El Objeto especial nil

```
>> [1, 2, 3][10]
=> nil
>> nil.to_s
=> “”
>> nil.to_i
=> 0
>> nil.object_id
=> 8

>> puts ‘hello’ unless nil
hello
```

# El Objeto especial nil

```
>> hash = { a: 1 }
>> hash[:b]
=> nil
>> hash[:b].nil?
=> true
```

# Excepciones en Ruby

# Excepciones en Ruby

```
>> 1 / 0
ZeroDivisionError: divided by 0
```

# Excepciones en Ruby

```
>> 1 / 10
ZeroDivisionError: divided by 0
```

# Excepciones en Ruby

- **RuntimeError**
  - `raise`
- **NoMethodError**
  - `Object.new.unknown_method!`
- **NameError**
  - `a = wrong_name`

# Excepciones en Ruby

- **IOError**
  - `STDIN.puts('Donr write here!')`
- **Errno::error**
  - `File.open(-12)`
- **TypeError**
  - `a = 3 + "can't add a string to a number!"`
- **ArgumentError**
  - `def m(x); end; m(1,2,3,4)`

# ¡**rescue** al rescate!

```
puts ‘enter a number’
number = gets.to_i

begin
  result = 10 / number
rescue
  puts “ERROR! Was your number zero?”
  exit
end

puts “10 / #{number} is #{result}”
```

# ¡**rescue** al rescate!

```
puts ‘enter a number’
number = gets.to_i

begin
  result = 10 / number
Rescue ZeroDivisionError
  puts “ERROR! Was your number zero?”
  exit
end

puts “10 / #{number} is #{result}”
```

# ¡**rescue** al rescate!

```
def my_method
  # some code
rescue
  puts “Couldn’t perform this action!”
end
```

# Lanzar excepciones

```
def fussy_method(x)
  raise ArgumentError, ‘Need a number under 10’ if x > 10
end
```

# ¡Asegurando con ensure!

```
begin
  # some code
rescue ArgumentError
  # rescue...
ensure
  # this code will always run!
end
```

# retry

```
tries = 3

begin
  # do something
rescue
  tries -= 1
  retry if > 0
end
```

# retry como un **rubista**

```
def try(n_times)
  yield
rescue Exception => e
  n_times -= 1
  if n_times > 0
    puts “Error #{e}. Retry!”
    retry
  end
end

>> try(3) { download_picture }
```

# Clases en Ruby

# Clases en Ruby

```
class Person
end

>> person = Person.new
>> person.class
=> Person
```

# Clases en Ruby

```
class Person
  def initialize(name, surname)
    @name = name
    @surname
  end
end

>> person = Person.new(‘John’, ‘Smith’)
```

# Clases en Ruby

```
class Person
  …
  def full_name
    “{@name} #{@surname}”
  end
end

>> person.full_name
=> John Smith
```

# Clases en Ruby

```
class Person
  ...
  def name
    @name
  end
end

>> person.name
=> John
```

# Clases en Ruby

```
class Person
  …
  def name=(new_name)
    @name = new_name
  end
end

>> person.name = ‘Will’
>> person.name
=> ‘Will’
```

# Clases en Ruby

```
class Person
  attr_accessor :name
end

>> person = Person.new
>> person.name = ‘Will’
>> person.name
=> ‘Will’
```

# Clases en Ruby - **getters** & **setters**

```ruby
class Person
  attr_reader :name     # .name returns @name (getter)
end

class Person
  attr_writer :name     # .name=() assigns @name (setter)
end

class Person
  attr_accessor :name   # both getter and setter
end
```

# Clases en Ruby - **encapsulación**

```
class Person
  def public_method
    # called from outside
    # example: Person.new.public_method
  end

  private

  def private_method
    # called from inside the class
  end
end
```

# Clases en Ruby - **encapsulación**

```ruby
class Person
  def public_method
    # called from outside
    # example: Person.new.public_method
  end

  protected

  def protected_method
    # called from inside the class
  end
end
```

# Clases en Ruby - **self**

```
class Person
  attr_accessor :name

  def foo(new_name)
    self.name = new_name
  end
end
```

# Herencia en Ruby

```ruby
class Publication
  attr_accessor :publisher
end

class Magazine < Publication
  attr_accessor :editor
end

class TheNews < Magazine
end
```

# Herencia en Ruby

```
class A
  def foo
    puts ‘A’
  end
end

class B < A
  def foo
    puts ‘B’
  end
end
```

# Herencia en Ruby

```
>> B.new.foo
B
class B < A
  def foo
    super
    puts ‘B’
  end
End

>> B.new.foo
A
B
```

# Herencia en Ruby

```
class B < A
  def my_method
    super(10)
  end
end
```

# Herencia en Ruby

```
class Person
  def initialize(name)
    @name = name
  end
End

class Worker < Person
  def initialize(name, job)
    super(name)
    @job = job
  end
end
```

# Métodos de clase

```
class MyClass
  def self.class_method
    puts ‘Hey!’
  end
end

>> MyClass.class_method
Hey!
>> MyClass.new.class_method
NoMethodError
```

# Métodos de clase

```
class Person
  @@count = 0

  def initialize(name)
    @name= name
    @@count += 1
  end

  def self.print_count
    puts “Person count is #{@@count}”
  end
end
```

# Métodos de clase

```
>> john = Person.new ‘John’
>> will = Person.new ‘Will’
>> Person.print_count
Person count is 2
```

# Constantes en Ruby

# Constantes en Ruby

```
A = “I’m a constant”

class Foo
  A = “I’m a constant of Foo class”
end

>> A
=> “I’m a constant”
>> Foo::A
=> “I’m a constant of Foo class”
```

# Modulos en Ruby

## Módulos son:

- Espacios de Nombre
- **Contenedores** de métodos
- Perfectos para **reutilizar** código

# Módulos en Ruby

```
module MyModule
  def module_method
    puts “module method!”
  end
end
```

# Módulos en Ruby

```
class A
  include MyModule
end

>> a = A.new
>> a.module_method
module method!
```

# Módulos en Ruby

```
class A
  extend MyModule
end

>> A.module_method
module method!
```

# Módulos en Ruby - **mixings**

```
module A
  # lots of methods
end

module B
  # more methods
end

class A
  include A
  include B
end
```

# Clases en Módulos

```
module MyModule
  class MyClass
  end
end

>> MyModule::MyClass
```

# Módulos - **ejemplo**

- module **Walkable**
- module **Swimmable**
- module **Climbable**
- clase **Animal**
  - **subclases**

# Let’s practice!

# Let’s practice - clase **Person**

- Atributos: nombre, apellido, edad
- Métodos:
  - Nombre Completo
  - Iniciales

# Let’s practice - clase **Dice**

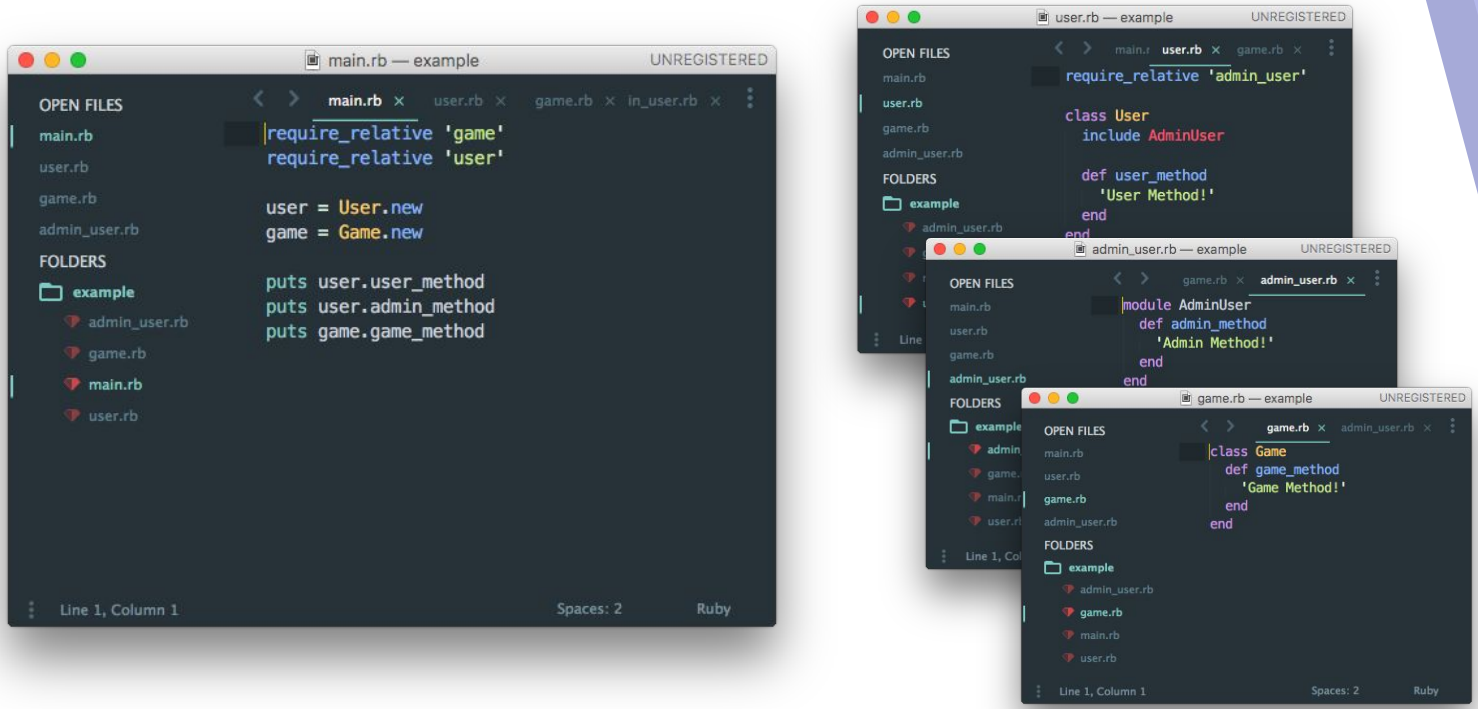
- Atributos: número de caras
- Métodos:
  - roll

# Organiza tu código

# Organiza tu código

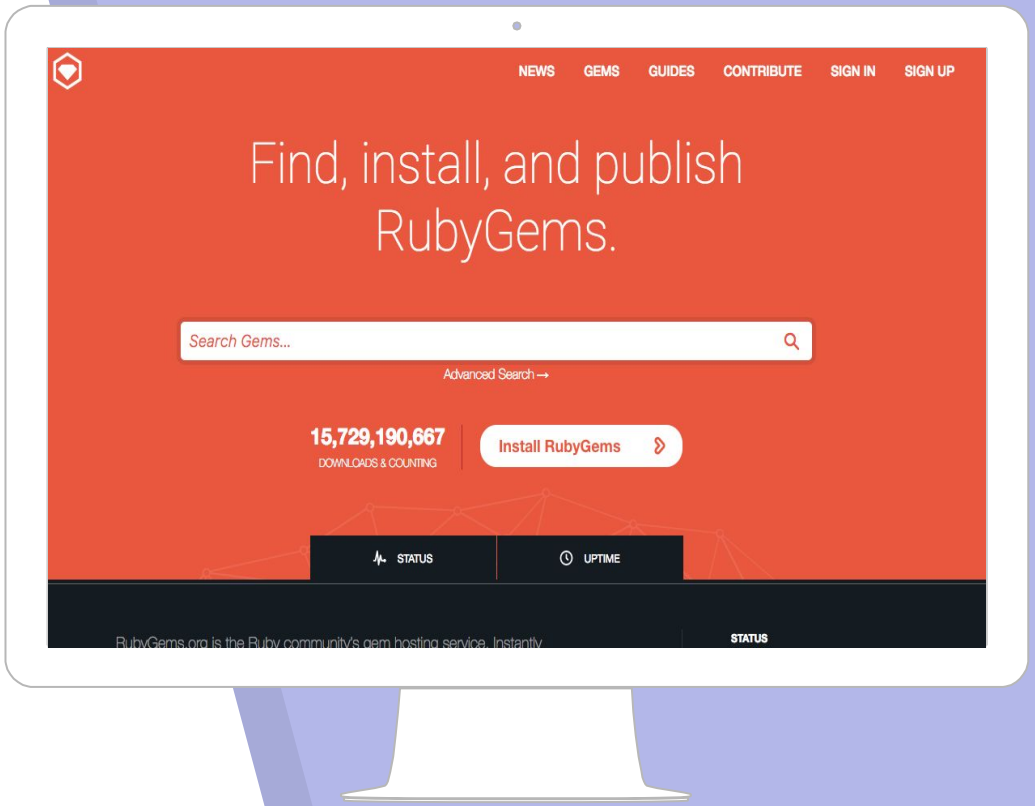
- Agrupa tus métodos en **módulos**
- ¡Métodos **pequeños**!
- Crea **clases**
- Aplica **herencia**
- Un **fichero** para cada clase/módulo

# Tu código en múltiples ficheros



# Ruby
# Gems

www.ruby
gems.org



# ¡Usando una gema!

```
$ gem install dni_nie
1 gem installed

$ irb
>> require ‘dni_nie’
>> DniNie.letra ‘45050505’
=> ‘Z’
>> DniNie.validate_doc '45050505Z'
=> true
>> DniNie.validate_doc '45050505K'
=> false
```

# ¡Usando una gema!

```
# ~/test.rb
require ‘dni_nie’

print ‘Enter your DNI number: ’
dni_num = gets.chomp
code = DniNie.letra(dni_num)
puts “That’s a #{code}!”
```

```
$ ruby ~/test.rb
Enter your DNI number: 49050505
That’s a S!
```

# Ruby debug con Pry

# ¡Debug con **pry**!

```
$ gem install pry

require ‘pry’

puts ‘some code’
a = :variable
binding.pry


pry>
```

# Ruby
# Bundler

# Ruby bundler

- Una **gema** para dominarlas a todas
- Las dependencias puede ser un **infierno**!
- Unirse a un proyecto es tan simple como **bundle install**

# Ruby bundler

```
$ gem install bundler


# Gemfile
source 'https://rubygems.org'
gem 'nokogiri'
gem 'rack', '~> 2.0.1'
gem 'rspec'


$ bundle install
$ git add Gemfile Gemfile.lock
```

# Gemfile

- Describe las **dependencias** de gemas
- En directorio **raíz** del proyecto
- Evaluado como **código** ruby
- ¡También versión de **ruby**!
- Grupos
- Fuentes
- Paths, git, ...

# The Gemfile - source

```
source "https://gems.example.com"


gem 'my_gem', '1.0', source: 'https://gems.example.com'


source 'https://gems.example.com' do
  gem 'my_gem', '1.0'
  gem 'another_gem', '1.2.1'
end
```

# The Gemfile - **versiones**

```
gem 'nokogiri'
gem 'rails', '5.0.0'
gem 'rack',  '>=1.0'
gem 'thin',  '~>1.1'    # >= 1.1, < 2.0
```

# The Gemfile - git / path

```
gem 'nokogiri', git: 'https://github.com/tenderlove/nokogiri.git',
                branch: '1.4'
```

```
gem 'extracted_library', path: './vendor/extracted_library'
```

# The Gemfile - **grupos**

```
gem 'wirble', :group => :development
gem 'debugger', :group => [:development, :test]

group :test do
  gem 'rspec'
end
```

# Gemfile.lock

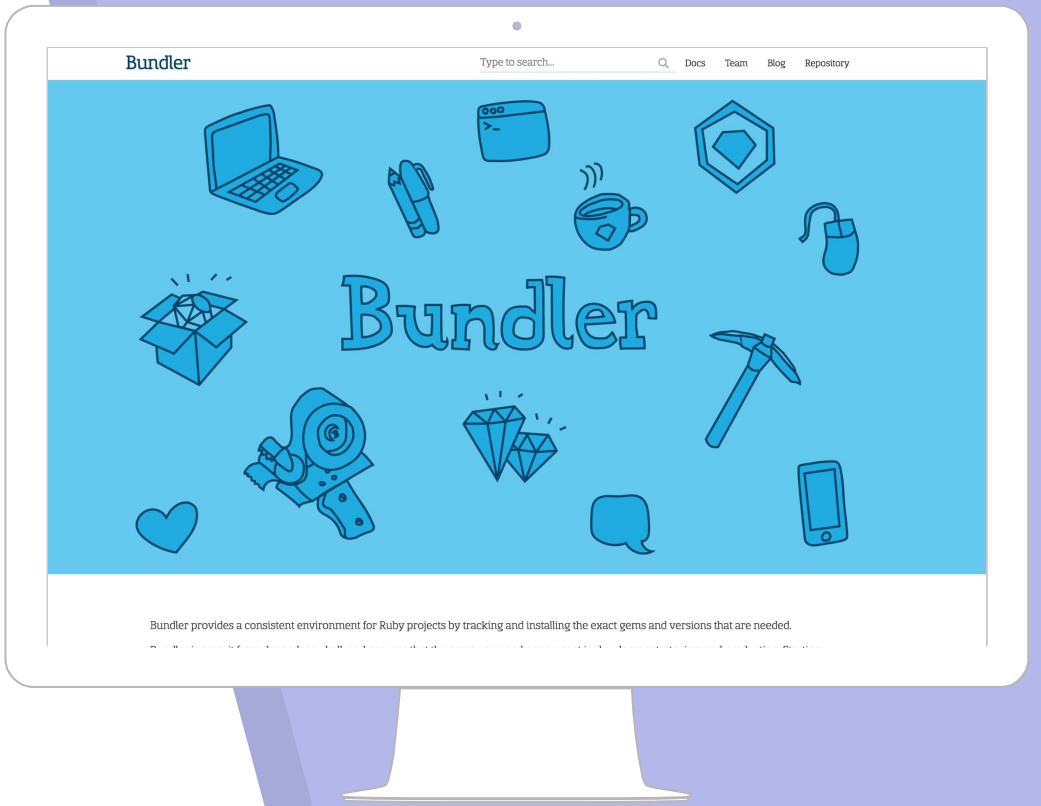
- Snapshot
- After bundle install
- ¡Gemas de **terceros**!

# Comandos bundle

- bundle **install**
- bundle **update**
- bundle **package**
- bundle **exec**

- bundle **gem**

# bundler.io


Bundler
Type to search...
Docs
Team
Blog
Repository
Bundler
Bundler provides a consistent environment for Ruby projects by tracking and installing the exact gems and versions that are needed.

# Ruby
## Metaprogramación

# Ruby es flexible y dinámico

- Para programadores **responsables**
- Permite **modificar** su core
- ¡Sin restricciones!

```ruby
class Numeric
  def plus(x)
    self.+(x)
  end
end

y = 5.plus 6
# y is now equal to 11
```

# Metaprogramación en Ruby

- **La idea**

  Usar código Ruby para **programar** Ruby dinámicamente

# Metaprogramación en Ruby - **object**

```
obj = Object.new

def obj.color=(color)
  @color = color
end

def obj.color
  @color
end

obj.color = :red
obj.color # => :red
Object.new.color # => NoMethodError
```

# Metaprogramación en Ruby
# ¿**mucho** código?

```ruby
class Task
  # statuses: new, in_progress, cancelled, resolved, feedback

  def new?
    @status == :new
  end

  def new!
    @status = :new
  end

  ...

end
```

# Metaprogramación en Ruby
# **method_missing**

```
def method_missing(name, *args, &block)
  …
  super # raises NoMethodError
end
```

# Metaprogramación en Ruby
## send

```
character = gets.chomp

puts game.send(“#{character}_abilities”)
```

# Ruby Testing con RSPEC

# Ruby Behaviour Driven Development

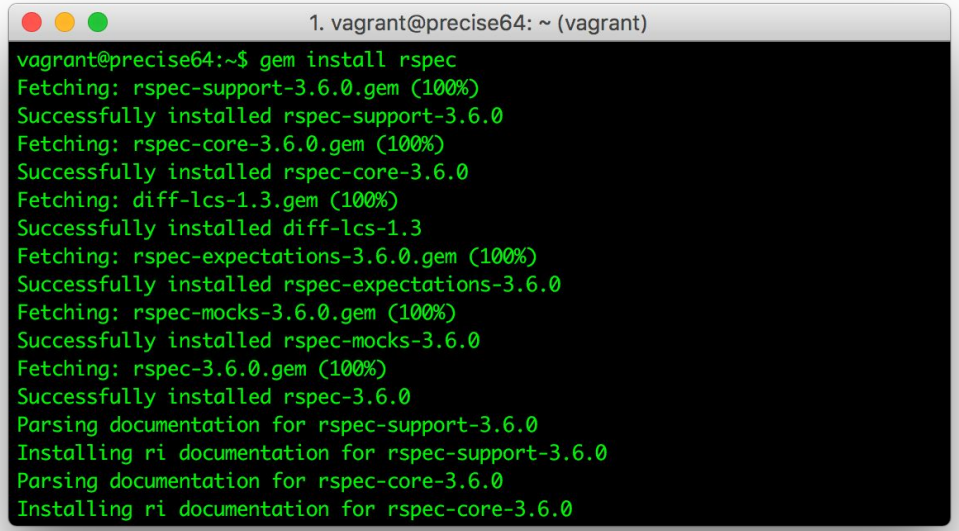
- Asegurar que tu código **funciona**
- Especificar el comportamiento
- ¡Productivo con **TDD**!
- ¡**Migrar** de versión es más fácil!

# ¿Qué es RSPEC?

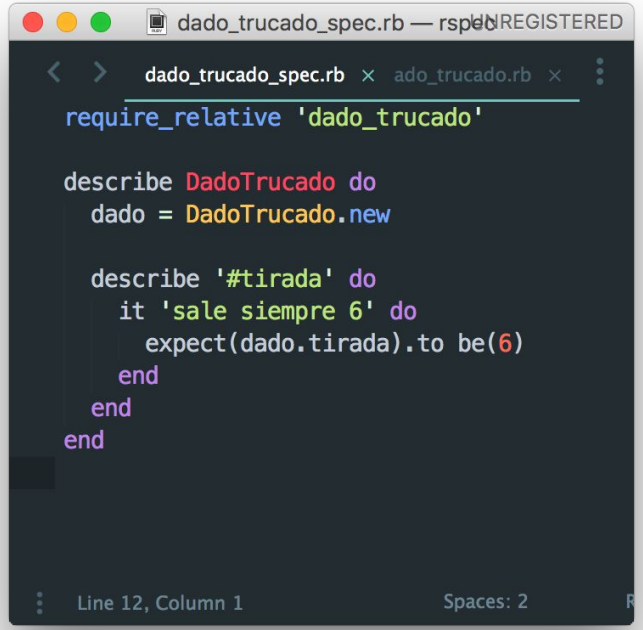
- ¡Una gema de Ruby!
  - rspec-core
  - rspec-expectations
  - rspec-mocks
  - rspec-rails

# Instalar rspec

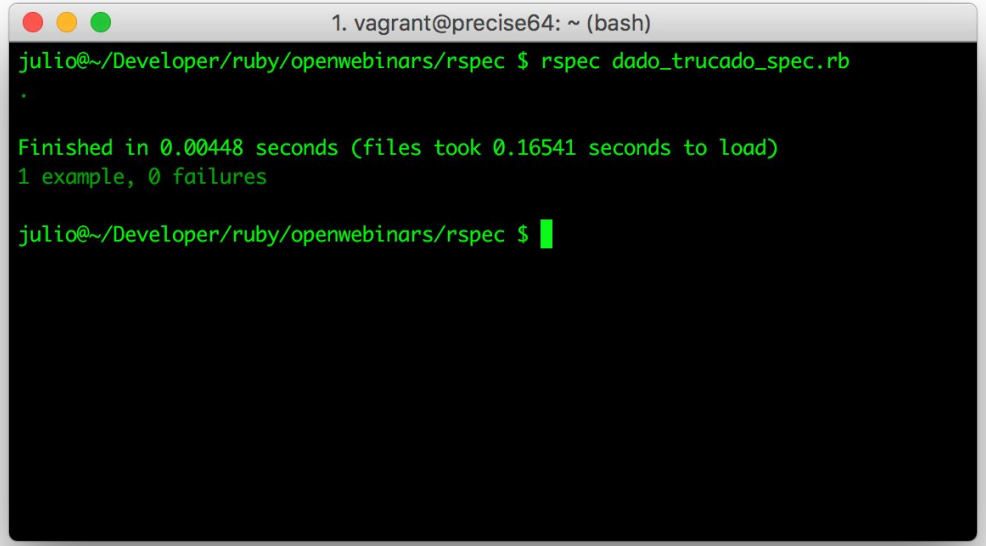
```
$ gem install rspec
```



# Estructura básica



# Ejecución


1. vagrant@precise64: ~ (bash)
julio@~/Developer/ruby/openwebinars/rspec $ rspec dado_trucado_spec.rb
.
Finished in 0.00448 seconds (files took 0.16541 seconds to load)
1 example, 0 failures
julio@~/Developer/ruby/openwebinars/rspec $

# Bloque **describe**

```
Rspec.describe MyClass do
  # rspec code
end
```

# **describes** anidados

```
Rspec.describe MyClass do
  describe ‘#instance_method’ do
    # tests
  end

  describe ‘.class_method’ do
    # tests
  end
end
```

# Bloque **context**

```
Rspec.describe MyClass do
  context ‘when given condition’ do
    # rspec  code
  end
end
```

# Bloque **it**

```
Rspec.describe MyClass do
  describe ‘#my_method’ do
    it ‘expectation message’ do
      # expect code
    end
  end
end
```

# Expectations

```
Rspec.describe MyClass do
  describe ‘#my_method’ do
    it ‘expectation message’ do
      expect(input).to eq(output)
    end
  end
end
```

# Expectations

```
expect(...).to
expect(...).not_to
```

# Expectations - **igualdad**

```
expect(a).to eq(b) # OK si a.eql?(b)
expect(a).to be(b) # OK si a.equal?(b)

expect(‘hello’).to eq(‘hello’)   # OK
expect(‘hello’).to be(‘hello’)   # ERROR

expect(:hello).to be(:hello)     # OK
expect(5).to be(5)               # OK
```

# Expectations - **comparación**

```
expect(9).to be > 6

expect(3).to be <= 3

expect(1).to be < 6

expect('a').to be < 'b'
```

# Expectations - **tipo**

```
expect(obj).to be_kind_of(type)
expect(obj).to be_a_kind_of(type)
expect(obj).to be_a(type)
expect(obj).to be_an(type)


expect(obj).to be_an_instance_of(type)
expect(obj).to be_instance_of(type)
```

# Expectations - **boolean**

```
expect(obj).to be_truthy    # not false or nil

expect(obj).to be_falsey    # false or nil

expect(obj).to be_nil       # nil
```

# Expectations - **change**

```
# Ahora expect recibe un bloque!

expect { Counter.increment }.to change { Counter.count }.from(0).to(1)

expect { Counter.increment }.to change { Counter.count }.by(1)
```

# www.rspec.info