

# Introducción a Ruby

Agile Lima Ruby Coding Dojo 24 de abril 2013

URL: <http://goo.gl/fom0A>

# ¿Para que sirve Ruby ?

- desarrollo web (Rails, Sinatra, etc.)
- herramientas de administración de sistema (Chef, Puppet)
- scripting

# Ruby es un lenguaje

- orientado a objetos:
  - *"david".reverse*
  - *"david".class*
- interpretado
- dinamico
- con sintaxis simple e intuitiva

# Sintaxis básica

- ; no es necesario al fin de una linea
- comentarios con #
- identificadores pueden tener a-zA-z, 0-9 y \_
  - no pueden empezar con \_

# Objetos

- casi todo es objeto un Ruby
- por defecto hereda de la clase *Object*
- no es necesario destruir los objetos (Ruby lo hace)
- se crea un objeto con *new*, se invoca el método *initialize* si existe

# Reserved words y objetos importantes

- *self* referencia al propio objeto
- *super* referencia al padre del objeto
- *nil* es un objeto (de clase *NilClass*) que designada el "nada"
- *true* y *false* son los booleanos
- todo evalua a *true* excepto *false* y *nil*

# Crear una class

```
class Man  
  def initialize(name) # invocado con "new"  
    @name = name  
  end  
  
  def hello  
    "my name is #{@name}!"  
  end  
end
```

# Usar una clase

```
joe = Man.new("joe")
```

```
joe.hello()
```

```
# => "my name is joe!"
```



# Otras cosas

- *ARGV* es un array con los parámetros que se pasan en línea de comandos
- *STDIN* es la entrada estándar y *STDIN.gets* lee hasta un final de línea
- *sleep(2)* hace esperar 2 segundos

# Prueba

- Un prueba es un código que prueba si un trozo de código (por ejemplo un metodo) funciona como se espera o no.
- Un prueba se escribe en Ruby, podemos tener varias pruebas en el mismo archivo.
- Se corre como código Ruby normal:
  - *ruby my\_test\_file.rb*

# Prueba (2)

- Usaremos la librería minispec
- con Ruby 1.9 ya es instalado
- con Ruby 1.8: *gem install minitest*
- Lista de las expectations de minispec: <http://www.ruby-doc.org/stdlib-1.9.3/libdoc/minitest/spec/rdoc/MiniTest/Expectations.html>

# Ejemplo de una prueba

```
require "minitest/autorun"
```

```
describe Man do
```

```
  it "#hello returns a sentence containing the  
  name" do
```

```
    man = Man.new("joe")
```

```
    man.hello().must_equal "my name is  
joe!"
```

```
  end
```

```
end
```

# Esqueleto para el kata

Puedan hacer un clone de

<https://github.com/Florent2/Ruby-Coding-Dojo-Boilerplate>

y consultar su README.md

kata: <https://github.com/12meses12katas/Diciembre-PomodoroKata>