



Disertante/s

Lic. Matias Andrés Beade

matiasbad@gmail.com

Lic. Jorge Adrián Chiavaro Alles

jorgechiavaro@gmail.com



Workshop Desarrollo Ágil con Ruby

Fundamentos de Ruby

Workshop Desarrollo Ágil con Ruby

Agenda de la charla

- Fundamentos
- Ruby básico
- Strings y simbolos
- Arrays y Hashes
- Estructuras de control
- Métodos y bloques
- Clases y objetos



Workshop Desarrollo Ágil con Ruby

Fundamentos

3 pilares de ruby:

- Todo es un objeto.
- Toda operación en un mensaje sobre un objeto que retorna un valor.
- Todo es “mata-programming”:
 - Todo puede ser modificado en tiempo de ejecución.
- Convenciones de codificación:
 - Varias formas de expresar lo mismo.
 - CamelCase y snake_case

Workshop Desarrollo Ágil con Ruby

Tema 2

Ruby básico:

- Sintaxis del lenguaje

Variable local	<code>name fish_and_chips x_axis thx1138 _x _26</code>
Variable de instancia	<code>@name @point_1 @X @_ @plan9</code>
Variable de clase	<code>@@total @@symtab @@N @@x_pos @@SINGLE</code>
Variable global	<code>\$debug \$CUSTOMER \$_ \$plan9 \$Global</code>
Nombre de clase	<code>String ActiveRecord MyClass</code>
Nombre de constante	<code>FEET_PER_MILE DEBUG</code>

<http://pastebin.com/zX1GnK8y>

Workshop Desarrollo Ágil con Ruby

Strings

Strings

- Secuencia de caracteres englobados entre “” o ‘’.
- Comillas simples vs. comillas dobles
- %q y %Q
- Métodos comunes

Simbolos

- Fundamentos y usos.
- Son manejados automáticamente por el lenguaje.

<http://pastebin.com/DcbmwabM>

<http://pastebin.com/PKk1cXqi>

<http://pastebin.com/p3sHsnpZ>

Workshop Desarrollo Ágil con Ruby

Arrays y Hashes

Arrays y Hashes

- Son colecciones indexadas y se acceden por clave.
- Son estructuras dinámicas.
- Arrays son accedidos por índice.
- Hashes son accedidos por clave.

<http://pastebin.com/z404SHzZ>

Workshop Desarrollo Ágil con Ruby

Estructuras de control

Estructuras de control

- if..else, case..when, unless
- Ruby loops
- brake, next, redo, retry

<http://pastebin.com/X6jKJg8F>

<http://pastebin.com/64V5X0HK>

<http://pastebin.com/XRngtyVR>

Workshop Desarrollo Ágil con Ruby

Métodos y Bloques

Métodos:

- Definición de métodos
- return
- Parámetros

<http://pastebin.com/CBLW7Vz4>

Bloques:

- Definición y sintaxis
- yield
- bloques y métodos

<http://pastebin.com/kvdy5Yv1>

Workshop Desarrollo Ágil con Ruby

Clases y objetos

Clases y objetos:

- Todo en ruby en un objeto.
- Los objetos son generados a partir de clases
- Constructores, métodos, atributos
- Relaciones entre clases
- Control de acceso

<http://pastebin.com/zVgikk6D>

Workshop Desarrollo Ágil con Ruby

Resumen

Variables	<code>local_variable</code> , <code>@@class_variable</code> , <code>@instance_variable</code>
Constants	<code>ClassName</code> , <code>CONSTANT</code> , <code>\$GLOBAL_CONSTANT</code>
Booleans	<code>false</code> , <code>nil</code> (es falso); <code>true</code>
Strings and symbols	<code>"string"</code> , <code>'other string'</code> , <code>%q{ comillas simples }</code> , <code>%Q{ comillas dobles }</code> , <code>:symbol</code>
Expressions	<code>@foo = 0</code> ; <code>"Answer is #{foo}"</code>
Regular expressions	<code>"hello" =~ /lo</code> or <code>"hello".match(Regexp.new 'lo')</code>
Arrays	<code>a = [1, :two, "three"]</code> ; <code>a[1] = "two"</code>
Hashes	<code>H = {:a => 1, 'b' => "two"}</code> ; <code>h['b'] == "two"</code> ; <code>h.has_key?(:a)</code>
Instance method	<code>def method(arg, arg) .. end</code> (*args for variable n° of params)
Class (static) method	<code>def ClassName.method(arg, arg) .. end</code> <code>def self.method(arg, arg) .. end</code>

Workshop Desarrollo Ágil con Ruby

Resumen

Variables	<code>local_variable</code> , <code>@@class_variable</code> , <code>@instance_variable</code>
Constants	<code>ClassName</code> , <code>CONSTANT</code> , <code>\$GLOBAL_CONSTANT</code>
Booleans	<code>false</code> , <code>nil</code> (es falso); <code>true</code>
Strings and symbols	<code>"string"</code> , <code>'other string'</code> , <code>%q{ comillas simples }</code> , <code>%Q{ comillas dobles }</code> , <code>:symbol</code>
Expressions	<code>@foo = 0</code> ; <code>"Answer is #{foo}"</code>
Regular expressions	<code>"hello" =~ /lo</code> or <code>"hello".match(Regexp.new 'lo')</code>
Arrays	<code>a = [1, :two, "three"]</code> ; <code>a[1] = "two"</code>
Hashes	<code>H = {:a => 1, 'b' => "two"}</code> ; <code>h['b'] == "two"</code> ; <code>h.has_key?(:a)</code>
Instance method	<code>def method(arg, arg) .. end</code> (*args for variable n° of params)
Class (static) method	<code>def ClassName.method(arg, arg) .. end</code> <code>def self.method(arg, arg) .. end</code>

Workshop Desarrollo Ágil con Ruby

Resumen

Special method names	<pre>def setter=(arg, arg) ... end def boolean_method?(arg, arg) ... end def dangerous_method!(arg, arg) ... end</pre>
----------------------	--

Conditionals	Iteration	Exceptions
<pre>if cond (or unless cond) statements [else if cond statements] [else statements] end</pre>	<pre>while cond (or until cond) statements end 1.upto(10) do i ... end 10.times do ... end collection.each do elt ... end</pre>	<pre>begin statements rescue AnError => e (e is an Exception of class AnError - multiple rescue is OK) [ensure this code is always executed] end</pre>

Workshop Desarrollo Ágil con Ruby

Referencias

- <http://www.ruby-doc.org/>
- <http://tryruby.org/levels/1/challenges/0>
- <http://www.ruby-lang.org/en/documentation/quickstart/>
- <http://www.ruby-doc.org/docs/ProgrammingRuby/>

Muchas gracias!!!



¿Dudas?

¿Consultas?