Ruby Tutorial

...o como pasar un buen rato programando

Lección 1

Lección 2

- Los métodos
- Los métodos: argumentos
- Rangos
- Arrays

Lección 3

- Bloques
- Más malabares con strings
- Expresiones Regulares
- Condicionales
- Bucles
- Números Aleatorios

Lección 4

- Clases y Objetos
- Accesores
- Ficheros: lectura/escritura
- Cargando librerías
- Herencia de clases
- Modificando clases
- Congelando objetos
- Serializando objetos

Lección 5

- Control de acceso
- Excepciones
- Módulos
- Constantes
- Hashes y Símbolos
- La clase Time

Lección 6

- self
- Duck Typing
- Azúcar Sintáctico
- Test de unidades

contacto

e-mail

Serializando Objetos

Java es capaz de **serializar objetos**: puede almacenarlos, para luego reusarlos cuando sea necesario. Ruby tiene también esta capacidad, pero la **llama **marshaling**.

Veamos un ejemplo en el que a partir de una clase, creamos una serie de objetos que almacenamos y luego recuperamos:

- La clase

personaje.rb

```
class Personaje
  def initialize(vida, tipo, armas)
    @vida = vida
    @tipo = tipo
```

```
    @armas = armas
  end
  attr_reader :vida, :tipo, :armas
end
```

- Creamos los objetos y los guardamos en un fichero usando **`Marshal.dump`**:

```
require 'personaje'
p1 = Personaje.new(120, 'Mago', ['hechizos', 'invisibilidad'])
puts p1.vida.to_s+' '+ p1.tipo+' '
p1.armas.each do |w|
  puts w + ' '
end

File.open('juego', 'w+') do |f|
  Marshal.dump(p1, f)
end
```

- Usamos **`Marshal.load`** para recuperarlos:

```
require 'personaje'
File.open('juego') do |f|
  @p1 = Marshal.load(f)
end

puts @p1.vida.to_s + ' ' + @p1.tipo + ' '
@p1.armas.each do |w|
  puts w + ' '
end
```

Marshal únicamente **serializa** estructuras de datos; no puede serializar código (como hacen los objetos `Proc`), o recursos utilizados por otros procesos (como conexiones a bases de datos). Marshal da un error cuando se intenta serializar un fichero.

Other interesting sites

Type Set Coin Collecting

zeromq

Scion: Monsters and Mosh Pits

Wiki for 'My Vineyard'

A Facebook Application