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

Más malabares con strings

Hay muchos métodos en la clase String (no hay que memorizarlos todos; para algo está la documentación) como:

- **`reverse`**, que invierte los caracteres de un string
- **`length`**, que nos dice el número de caracteres de un string, incluyendo los espacios en blanco.
- **`upcase`**, que pone todos los caracteres en mayúsculas
- **`lowercase`**, que pone todos los caracteres en minúsculas
- **`swapcase`**, pone los caracteres mayúsculas en minúsculas y los minúsculas en mayúsculas
- **`capitalize`**, pone el primer caracter de cada palabra en mayúsculas, y los demás en minúsculas
- **`slice`**, da una parte de un string

Los métodos `upcase`, `downcase`, `swapcase` y `capitalize` tienen su correspondiente método bang, que modifican el string (`upcase!`, `downcase!`, `swapcase!`, y `captalize!`). Si no

necesitas el string original, es bueno usarlo, por que salvarás memoria; sobretodo si el string es largo.

Cada vez que se se asigna a una variable un string, un nuevo objeto String es creado. ¿Cómo es administrada la memoria en los strings? ¿Hay una porción separada para ellos? La clase String tiene más de 75 métodos. Leyendo la Guía de Uso de Ruby (Ruby User's Guide), dice "no tenemos en cuenta la memoria ocupada por un string. Prescindimos de cualquier administración de memoria". Para saber todos los métodos que tiene un String:

- **`String.methods`**, da una lista de todo los métodos que tiene la clase String.
- **`String.methods.sort`** (sort=ordenar), da una lista ordenada alfabéticamente de todos los métodos.
- **`String.instance_methods.sort`**, da una lista ordenada de todo los métodos de instancia (se explicará más adelante) que tiene un String.
- **`String.instance_methods(false).sort`**, muestra una lista ordenada de todos los métodos que pertanezcan exclusivamente a los Strings, pero no a las clases de las que desciende.

Comparando dos cadenas

Los strings tienen distintos métodos para comparar su igualdad. El más común de ellos es **`==`**. Otro método es **`String.eql?`**, que devuelve el mismo resultado que ==. Y por último está **`String.equal?`**, que comprueba si dos strings son el mismo objeto. Veamos el siguiente ejemplo:

```
def compara_strings(s1, s2, s3)

#comprobamos si el contenido es igual
   if s1 == s2
     puts 'Ambos strings tienen el mismo contenido'
   else
     puts 'Ambos strings NO tienen el mismo conenido'
   end

   if s1.eql?(s2)
     puts 'Ambos strings tienen el mismo contenido'
   else
     puts 'Ambos strings NO tienen el mismo conenido'
   end

=begin
  ahora comprobamos si ambos objetos son iguales:
  dos objetos diferentes
  pueden tener el mismo contenido
=end
```

```
   if s1.equal?(s2)
     puts 'Ambos strings son el mismo objeto'
   else
     puts 'Ambos strings NO son el mismo objeto'
   end

   if s1.equal?(s3)
     puts 'Ambos strings son el mismo objeto'
   else
     puts 'Ambos strings NO son el mismo objeto'
   end

end

 string1 = 'Jonathan'
 string2 = 'Jonathan'
 string3 = string1

compara_strings(string1,string2,string3)
```

Ejercicio

Dado un string, invertirlo palabra por palabra (en vez de letra por letra).

Solución
Se puede usar **`String.split`** que nos da un array formado por las palabras del string. La clase Array tiene un método **`reverse`**; de tal forma que puedes revertir el array antes de juntarlo para hacer un nuevo string:

```
palabras = 'Tutorial de Ruby - fácil, sencillo y con fundamento'
puts palabras.split(" ").reverse.join(" ")
```

Other interesting sites

••••••

••••••••••

Let The End Times Roll

Harry Potter: Into The Fire

WWW 6

Wakacyjne Warsztaty Wielodyscyplinarne 6