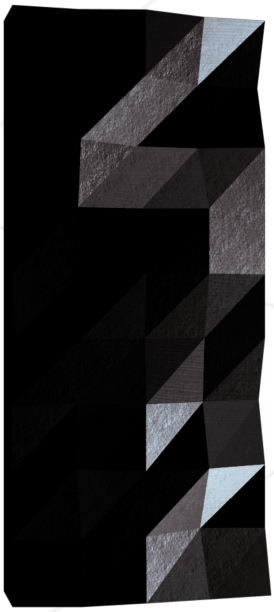

Ruby - Unidad 06

Métodos y scope

Staff Pedagógico 42 pedago@42.es

Resumen: En esta unidad vamos a ver cómo utilizar los métodos y el ámbito o alcance (scope).

Índice general

I.	**Preámbulo**	**2**
II.	**Instrucciones Generales**	**4**
III.	**Ejercicio 00: hello_all**	**5**
IV.	**Ejercicio 01: upcase_it**	**6**
V.	**Ejercicio 02: downcase_all**	**7**
VI.	**Ejercicio 03: greetings_for_all**	**8**
VII.	**Ejercicio 04: methods_everywhere**	**10**
VIII.	**Ejercicio 05: scope_that**	**12**

Capítulo I

Preámbulo

`Receta del kouign-amann`

Preparación: 4h 00 min
Cocción: 35 minutes

Ingredientes

250 g de harina
200 g de mantequilla
200 g de azúcar
10 g de levadura fresca
10 cl de agua
2 pizcas de sal

Instrucciones

Piense en sacar la mantequilla semisalada o salada* del frigorífico para que se vaya ablandando.
Empezamos haciendo la masa del kouign-amann. Mezclamos la levadura de panadería fresca (¡sobre todo que no sea levadura química!) con 3 cucharadas soperas de agua templada que no queme en una taza. Después, en un bol mezclamos la harina de trigo y añadimos 2 pizcas de sal (cuidado, la sal y la levadura de panadería no tienen que estar en contacto, de lo contrario se podría matar la levadura y no subiría la masa).
Hacemos un pozo y vertemos dentro la mezcla de la levadura con los 10 cl de agua.
Enharinamos la superficie de trabajo y amasamos la masa del kouign-amann hasta que quede elástica. Hacemos una bola con la masa y la dejamos reposar en el bol a temperatura ambiente durante 3 h (etapa 1 del esquema).
Al cabo de 3 h de reposo, la masa habrá triplicado su volumen. En la superficie de trabajo enharinada, estiramos la masa con un rodillo de repostería (esta etapa puede ser bastante larga ya que la masa está relativamente elástica) y le damos forma de rectángulo o de cuadrado de aproximadamente 1 cm de alto (etapa 2 del esquema). Untamos por encima la mantequilla semisalada bien blanda con un pincel y espolvoreamos azúcar. Cuidado con no echar mantequilla y azúcar en los bordes de la masa y con dejar un espacio de 3 cm en todo el borde (etapa 3 del esquema).
Doblamos la masa del kouign-amann en 3 a lo largo (etapa 4 del esquema) y de nuevo

en 3 a lo ancho (etapa 5 del esquema), como si fuese un hojaldre, el objetivo es que la mantequilla y el azúcar queden bien .[en]cerradas.[en] la masa.
Con el rodillo de repostería, volvemos a estirar la masa con mucha delicadeza para que no se salga la mantequilla (etapa 6 del esquema).
Volvemos a doblar la masa en 3 a lo largo y a lo ancho (etapas 7 y 8 del esquema).
Introducimos la masa en un molde embadurnado con mantequilla, la comprimimos y la dejamos reposar media hora.
Con un cuchillo, dibujamos una cuadrícula sobre la masa y esparcimos algunos trozos más de mantequilla.
Precalentamos el horno a 210 C^{o} (termostato 7).
Después, horneamos el kouign-amann durante unos 35 mn. Cuando el kouign-amann esté cocido, lo sacamos del horno y esperamos un cuarto de hora antes de sacarlo del molde.
Espolvoreamos el kouign-amann con un poco de azúcar y lo degustamos mientras que todavía está templado.

Truco del chef:
A raíz de un comentario de un internauta, no le recomiendo que utilice mantequilla con cristales de sal de Guérande o con sal gorda. Los granos de sal gorda tienden a romper la masa cuando se unta la mantequilla y hacen que esta última se escape durante la cocción.

Capítulo II

Instrucciones Generales

Salvo que se indique lo contrario de forma explícita, las siguientes instrucciones serán válidas durante el tiempo que dure la Piscina.

- Esta evaluación será la única referencia: no se fíe de los rumores de pasillo.
- Los ejercicios han sido ordenados con mucha precisión del más sencillo al más complejo. En ningún caso prestaremos atención a un ejercicio complejo ni lo tendremos en cuenta si no se ha conseguido realizar perfectamente un ejercicio más sencillo.
- Tenga cuidado con los permisos de sus archivos y de sus directorios.
- Sus compañeros de Piscina se encargarán de corregir los ejercicios que usted realice.
- Para la evaluación entre pares, no debe dejar en su directorio ningún archivo que no haya sido indicado de forma explícita en los enunciados de los ejercicios.
- ¿Tiene alguna pregunta? Pregunte a su vecino de la derecha. Si no, pruebe con su vecino de la izquierda.
- Todas las respuestas a sus preguntas técnicas se encuentran en los `man` o en Internet.
- ¡No olvide participar en el foro Piscina de su Intranet y en el Slack!
- Lea detenidamente los ejemplos ya que pueden ayudarle a identificar algún trabajo que tenga que realizar y que, a primera vista, no venga explicado en el enunciado.
- Razone. ¡Se lo suplico, por Thor, por Odín!

Capítulo III

Ejercicio 00: hello_all

	Ejercicio : 00
¡Descubrimos los métodos!	
Directorio de entrega : $ex00/$	
Ficheros a entregar : `hello_all.rb`	
Funciones autorizadas : `Todas`	

- Cree el programa `hello_all.rb`.
- Tiene que ser un programa ejecutable.
- Este programa debe incluir el método hello. Este método tiene que mostrar "¡Hola a todos!".
- Una vez definido el método, pruébelo llamándolo dentro del programa. Como en el siguiente ejemplo, salvo que hemos escondido la definición del método.

```
?> cat hello_all.rb
#su definición de método

hello()
?> ./hello_all.rb
¡Hola a todos!
?>
```

Busque la "definición de un método en Ruby".

Capítulo IV

Ejercicio 01: upcase_it

	Ejercicio : 01
!El retorno de upcase!	
Directorio de entrega : $ex01/$	
Ficheros a entregar : `upcase_it.rb`	
Funciones autorizadas : `Todas`	

- Cree el programa `upcase_it.rb` (¿otra vez?).
- Tiene que ser un programa ejecutable.
- Debe definir un método en ese programa. El método se llamará `upcase_it`.
- El método `upcase_it` tiene que recibir como argumento una cadena de caracteres. Tendrá que devolver esa cadena de caracteres en mayúsculas.
- Pruebe el método llamándolo dentro del programa. En el siguiente ejemplo hacemos una prueba con çucú":

```
?> cat upcase_it.rb
# su definición de método

puts upcase_it("coucou")
?> ./upcase_it.rb
CUCÚ
?>
```

Capítulo V

Ejercicio 02: downcase_all

	Ejercicio : 02
Paseémonos por el array	
Directorio de entrega : $ex02/$	
Ficheros a entregar : `downcase_all.rb`	
Funciones autorizadas : `Todas`	

- Cree el programa `downcase_all.rb`.
- Tiene que ser un programa ejecutable.
- Debe definir un método en ese programa. El método se llamará `downcase_it`.
- El método `downcase_it` tendrá que recibir como argumento una cadena de caracteres. Después, devolverá esa cadena de caracteres en minúsculas.
- Ejecutará el método y mostrará su retorno con los parámetros del programa.
- Si no hay ningún parámetro, muestre `none` seguido de un salto de línea.

```
?> ./downcase_all.rb
none
?> ./downcase_all.rb "HELLO WORLD" "¡Qué bien, he entendido las tablas!"
hello world
¡Qué bien, he entendido las tablas!
?>
```

Capítulo VI

Ejercicio 03: greetings_for_all

	Ejercicio : 03
¡Salude a la señora!	
Directorio de entrega : $ex03/$	
Ficheros a entregar : `greetings_for_all.rb`	
Funciones autorizadas : `Todas`	

- Cree el programa `greetings_for_all.rb` que no reciba parámetros.
- Tiene que ser un programa ejecutable.
- Dentro del programa, cree el método `greetings` que reciba un nombre como parámetro y muestre un mensaje de bienvenida con ese nombre.
- Si se llama al método sin argumento, el parámetro por defecto será "noble desconocida".
- Si se llama al método con un argumento que no sea una cadena de caracteres, se mostrará un mensaje de error en lugar del mensaje de bienvenida.
- De ese modo, el programa siguiente:

```
?> cat greetings_for_all.rb | cat -e
# su definición de método

greetings('Lucía')
greetings()
greetings(22)
```

tendrá como salida:

```
?> ./greetings_for_all.rb | cat -e
Hello, Lucía.$
Hello, noble desconocida.$
¡Error! No es un nombre.$
?>
```

`Google parámetro por defecto, método is_a.`

Capítulo VII

Ejercicio 04: methods_everywhere

	Ejercicio : 04
¡Métodos por todas partes!	
Directorio de entrega : $ex04/$	
Ficheros a entregar : `methods_everywhere.rb`	
Funciones autorizadas : `Todas`	

- Cree el programa `methods_everywhere.rb` que reciba parámetros.
- Tiene que ser un programa ejecutable.
- Dentro del programa, cree dos métodos distintos:
- El método `reduce` recibirá como parámetro una cadena de caracteres y tendrá que mostrar los ocho primeros caracteres de la cadena.

`Utilice los slices.`

- El método `agranda` recibirá como parámetro una cadena de caracteres y tendrá que completarla con "Z"para que en total haya ocho caracteres. Después tendrá que mostrar la cadena.

`Como en los arrays, podemos añadir caracteres a una cadena con el operador «`

- Para cada argumento del programa: si el argumento tiene más de 8 caracteres, llame al método `reduce`; si el argumento tiene menos de 8 caracteres, llame al método

`agranda`; y si el argumento tiene 8 caracteres, muéstrelo directamente seguido de un salto de línea.

- Si el número de parámetros es inferior a 1, se mostrará `'none'` seguido de un salto de línea.

```
?> ./methods_everywhere.rb | cat -e
none$
?> ./methods_everywhere.rb 'lol' 'agradablemente' 'valiente' | cat -e
lolZZZZZ$
agradabl$
valiente$
?>
```

Capítulo VIII

Ejercicio 05: scope_that

	Ejercicio : 05
Can't touch this!	
Directorio de entrega : $ex05/$	
Ficheros a entregar : `scope_that.rb`	
Funciones autorizadas : `Todas`	

- Cree el programa `scope_that.rb` que no reciba parámetros.
- Tiene que ser un programa ejecutable.
- Dentro del programa, cree el método `add_one` que reciba un parámetro y añada 1 al parámetro pasado al método.
- Inicialice une variable en el cuerpo del programa, muéstrela y llame al método que añade 1.
- Muestre de nuevo la variable en el cuerpo del programa.
- ¿Qué observa?