

Programación para todos

4. Arrays & Hashes



le wagon

@lewagonargentina

Arrays

una colección (o lista) de datos de Ruby que almacena una lista de valores (llamados elementos) en una sola variable

arrays pueden contener: **números** (en cualquier orden repetidos o sin repetir), **strings**, **booleans**, **symbols** y también **otros arrays!** :) (arrays de arrays se llaman *arrays multidimensionales*)

arrays están definidos por valores específicos entre square brackets [], separados por comas

```
my_array = ["Bob", "Joe", "Zack"]
```

Arrays (cont.)

cada elemento en el array tiene un **índice** - > el primer elemento es índice **0**, el siguiente índice 1, el siguiente índice 2, etc.

podemos acceder a los elementos de cualquier array colocando el índice entre **square brackets**

```
array = [5, 7, 9, 2, 0]  
array[1]
```

(returns "7", porque "7" se encuentra en el índice 1)

Arrays (cont.)

podemos agregar (**add**) elementos a un array

`.push(nuevo elemento)` o `<< nuevo elemento`

podemos eliminar (**delete**) elementos de un array

`.delete_at(index)`

Hashes

una colección de datos de Ruby que almacena una lista de parejas **key-value** (clave - valor)
en una única variable podemos usar cualquier **objeto** de Ruby como key o value

values son asignados a los keys usando un **rocket =>**

```
hash_name = {  
  key1 => value1,  
  key2 => value2,  
  key3 => value3  
}
```

Hashes (cont.)

creando un nuevo hash:

```
mi_hash = {  
  "gato" => "Garfield",  
  "perro" => "Snoopy"  
}
```

o

```
mi_hash = Hash.new  
mi_hash["gato"] =  
"Garfield"  
mi_hash["perro"] = "Snoopy"
```

Hashes (cont.)

podemos acceder (**access**) a una específico key-value así:

```
mi_hash = {  
  "gato" => "Garfield",  
  "perro" => "Snoopy"  
}  
puts mi_hash["gato"] (va a imprimir)
```

"Garfield"

Hashes (cont.)

podemos agregar (**add**) un key-value a un hash

```
mi_hash = {  
  "cat" => "Garfield",  
  "dog" => "Snoopy"  
}  
mi_hash["raton"] = "Mickey"
```

podemos eliminar (**delete**) un key-value

de un hash `mi_hash.delete("perro")`

Iterando... devuelta!

podemos hacer un loop sobre un array o hash, y podemos decir que estamos iterando sobre ellos

Iterando...devuelta

1. Iterando sobre un Array

```
my_array = ["Bob", "Joe", "Zack"]
```

```
my_array.each do | names |
```

```
  puts names
```

```
end
```

```
my_array.each { | names | puts names }
```

(ambos van a imprimir Bob, Joe, Zack)

Iterando...devuelta

2. Iterando sobre un Array Multidimensional

```
my_array = [["Bob", "Joe", "Zack"], ["Zoe", "Nina", "Chloe"]]
```

```
my_array.each do | sub_array |
```

```
  sub_array.each do | names |
```

```
    puts names
```

```
  end
```

```
end
```

(imprimirán Bob, Joe, Zack, Zoe, Nina, Chloe)

Iterando...devuelta

3. Iterando sobre un Hash

necesitamos dos **placeholders** para representar cada key/value

```
students_grades = {  
  "Zack" => 7,  
  "Zoe" => 10  
}
```

```
students_grades.each do |student, grade|  
  puts "#{student}: #{grade}"  
end  
(imprimirá Zack: 7, Zoe: 10)
```

Gracias !



le wagon

@lewagonargentina