

# Programación para todos

## 6. Hashes & Symbols



le wagon

@lewagonargentina

# Arrays vs Hashes

**Las colecciones** almacenan una lista de elementos en una variable única y pueden contener cualquier tipo de data (números, strings, booleans, otros arrays, otros objects, etc.)

**Arrays** son colecciones simples, donde se puede acceder a cada elemento solo por su posición (llamada index)

**Hashes** son colecciones más complejas, donde cada elemento es etiquetado con una *keyword*, y se puede acceder con esa palabra específica (la key)

# Arrays vs Hashes

# Arrays

```
my_array = [6, "a string", true, ["inside a sub-array!", 4, false], -12]
```

## indexes

0

1

2

3

4

# Hashes

# my\_hash = {

keys


"cat" => "Garfield",

"dog" => "Snoopy",

"mouse" => "Mickey"

}

# Symbols

Como vimos hasta ahora, strings se pueden usar como keys de nuestros hashes. 

Pero hay una mejor forma y más específica para representar los keys de un hash: **symbols**

**Recuerden siempre que un symbol es diferente de un string!**

Symbols pueden estar afuera del hash, y pueden ser asignadas a variables y usadas como cualquier otro tipo de data:

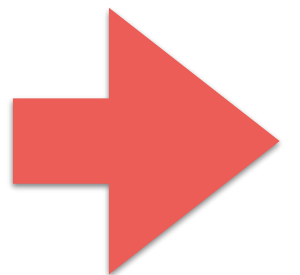
```
my_symbol = :this_is_a_symbol
```

# Symbols como Hash Keys

De ahora en más, cuando estemos tratando con hashes, intentemos de no usar *strings como keys*

Siempre tratemos de usar **symbols**, dado que son los más usados como keys y Ruby mejora su performance!

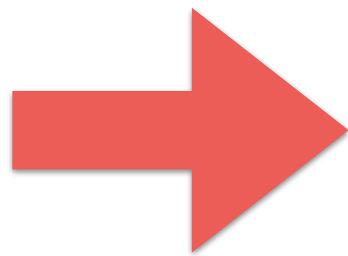
También, solo coloca dos puntos después (en un hash) o antes (afuera del hash) del symbols



# Symbols como Hash Keys

```
my_hash= {  
  "cat" =>  
    "Garfield", "dog"  
=> "Snoopy",  
  "more words" => "a  
value"  
}
```

```
cat_name= my_hash["cat"]
```



```
my_hash = {  
  cat:  
    "Garfield",  
  dog: "Snoopy",  
  more_words: "a  
value"  
}
```

```
cat_name= my_hash[:cat]
```

## No more rocket syntax!

# Gracias!



le wagon

@lewagonargentina