

[Wikidot.com](http://Wikidot.com)

.wikidot.com

Share on      

[Edit](#) [History](#) [Tags](#) [Source](#)

[Explore »](#)

# Ruby Tutorial

**...o como pasar un buen rato programando**

- [admin](#)
  - [site manager](#)

[Create account](#) or [Sign in](#)



## Lección 1

- [Introducción](#)
- [Instalación](#)
- [El Primer Programa](#)
- [Números en Ruby](#)
- [Strings y diversión](#)
- [Variables](#)
- [Alcance de las variables](#)

## Lección 2

- [Introduciendo Datos](#)
- [Normas en los nombres](#)
- [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](#)



## Hashes y Símbolos

Un símbolo parece una variable, pero está precedido de dos puntos. Ejemplos:

```
:action  
:line_tines
```

Los dos puntos se pueden interpretar como "la cosa llamada". Entonces `:id`, se interpreta como "la cosa llamada id". Los **símbolos** no contienen valores como las variables. Un símbolo es una etiqueta, un nombre, nada más.

## Símbolos vs Strings

Un **símbolo** es el objeto más básico que puedes crear en Ruby: es un nombre y una ID interna. Los símbolos son útiles por que dado un símbolo, se refiere al mismo objeto en todo el programa. Por lo tanto, son más eficientes que los strings: dos strings con el mismo nombre, son dos objetos distintos. Esto implica un ahorro de tiempo y memoria.

```
puts "hola".object_id    # 21066960  
puts "hola".object_id    # 21066730  
puts :hola.object_id     # 132178  
puts :hola.object_id     # 132178
```

Cada vez que se ha usado un string, se ha creado un objeto nuevo. Por tanto, ¿cuándo usar un string, y cuándo un símbolo?

- Si el contenido del objeto es lo importante, usa un string.
- Si la identidad del objeto es importante, usa un símbolo.

Ruby usa una tabla de símbolos interna con los nombres de las variables, objetos, métodos, clases... Por ejemplo, si hay un método con el nombre de `control_movie`, automáticamente se crea el símbolo `:control_movie`. Para ver la tabla de símbolos `Symbol.all_symbols`.

Como veremos a continuación, los símbolos son particularmente útiles para los hashes.

# Hashes

**Hashes**, también conocidos como arrays asociativos, mapas o diccionarios, son parecidos a los arrays en que son una colección indexada de referencias a objetos. Sin embargo, mientras que en los arrays los índices son números, en los hashes se puede indexar con objetos de cualquier tipo: strings, expresiones regulares, etc.

Cuando se almacena un valor en un array, se dan dos objetos: el índice y el valor. A posteriori se puede obtener dicho valor, gracias al índice.

```
h = {'perro' => 'canino', 'gato' => 'felino', 'burro' => 'asno', 12 => 'docena'}
puts h.length      # 4
puts h['perro']    # 'canino'
puts h
puts h[12]
```

Comparados con los arrays, tenemos una ventaja significativa: se puede usar cualquier objeto como índice. Sin embargo, sus elementos no están ordenados, y es fácil usar un hash como una pila o cola.

Los hashes tienen un valor por defecto. Este valor se devuelve cuando se usan índices que no existen: el valor que se devuelve por defecto es **nil**.

La clase Hash tiene muchos métodos que se pueden ver [aquí](#).

## Los símbolos como índices

Por las ventajas antes citadas, se usan los símbolos como índices:

```
persona = Hash.new
persona[:nombre] = 'Pedro'
persona[:apellido] = 'Picapiedra'

puts persona[:nombre]
```

que es equivalente a:

```
persona = {:nombre => 'Pedro', :apellido => 'Picapiedra'}

puts persona[:apellido]
```

Page tags: [params](#)

page\_revision: 5, last\_edited: 27 Jan 2009, 17:42 GMT-06 (671 days ago)

[EditTags](#) [History](#) [Files](#) [Print](#) [Site tools+](#) [Options](#)

[Help](#) | [Terms of Service](#) | [Privacy](#) | [Report a bug](#) | [Flag as objectionable](#)

Powered by [Wikidot.com](#)

Unless otherwise stated, the content of this page is licensed under [Creative Commons Attribution-ShareAlike 3.0 License](#)

## Other interesting sites



## [Gender in Media](#)



## [Portland ImmunoResearch Group](#)



## [Managing Information Systems](#)

BUA 581 - Rick Sprenkle



## RetroWiki