

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](#)



Arrays

Un **array** (o lista) es un conjunto ordenado: cada posición en la lista es una variable que podemos leer y/o escribir.

```
# Arrays (o vectores)
```

```
# array vacío  
vec1 = []
```

```
# Los índices empiezan desde el cero (0,1,2,...)  
nombres = ['Satish', 'Talim', 'Ruby', 'Java']  
puts nombres[0]  
puts nombres[1]  
puts nombres[2]  
puts nombres[3]
```

```
# si el elemento no existe, se devuelve nil
puts nombres[4]
# pero podemos añadir a posteriori más elementos
nombres[3] = 'Python'
nombres[4] = 'Rails'
```

Un array puede ser un conjunto de elementos distintos:

```
=begin
  un array cuyos elementos
  apuntan a otros tres objetos:
  un decimal, un string y un array
=end
sabor = 'mango'
vec4 = [80.5, sabor, [true, false]]
puts vec4[2]
```

Usando %w

Algunas veces, crear arrays de palabras puede ser tedioso debido a tantas comillas y comas. Afortunadamente, Ruby tiene una forma más cómoda para hacerlo:

```
nombres1 = [ 'ann', 'richard', 'william', 'susan', 'pat' ]
puts nombres1[0] # ann
puts nombres1[3] # susan
```

```
# esto es más sencillo y más rápido:
nombres2 = %w{ ann richard william susan pat }
puts nombres2[0] # ann
puts nombres2[3] # susan
```

El método each

El método **each** extrae cada elemento del array dentro de la variable que se le especifique (que irá entre dos barras | |), que se usará en bloque do...end.

```
ciudades = %w{ Pune Mumbai Bangalore }
ciudades.each do |ciudad|
  puts '¡Me gusta ' + ciudad + '!'
  puts '¿A ti no?'
end
```

```
#El método {**delete**} borra un elemento
ciudades.delete('Mumbai')
ciudades.each do |ciudad|
  puts '¡Me gustaba '+ciudad+'!'
  puts '¿A ti ya no?'
end
```

Por lo tanto el método `each` nos permite hacer una cosa (la que sea) con cada objeto del array. En el ejemplo, fuimos elemento por elemento del array sin usar los índices. Hay que destacar:

- Los variable entre los "postes" se refiere a cada ítem del array a medida que avanzamos en el loop. Se puede usar cualquier nombre, pero es mejor dotarlo de cierto significado.
- El **do...end** identifica el bloque de código que se ejecutará con cada elemento del array. Los bloques son usados intensivamente en Ruby, y se tratarán en profundidad más adelante.

Otros métodos

```
vec = [34, 12, 1, 38]
puts vec.sort
puts vec.length
puts vec.first
puts vec.last
```

Obteniendo arrays

Un método puede devolver un array:

```
def num_cuadrado(num)
  cuadrado = num * num
  return num, cuadrado
end

=begin
  el método nos devuelve
  un array con el número
  y su cuadrado
=end

x=3
num_cuadrado(x)
```

```
=begin
  si queremos almacenar el resultado
  hay que hacerlo por
  asignación en paralelo
=end

num, cuadrado = num_cuadrado(x)
```

Ejercicios

1. Escribe un programa tal que, dado un array numérico, calcule la suma de sus elementos. Por ejemplo, array = [1, 2, 3, 4, 5]
2. Escribe un programa tal que, dado un array de números, diga de cada uno si es par o impar. Por ejemplo, array = [12, 23, 456, 123, 4579]

page_revision: 7, last_edited: 27 Jan 2009, 10:29 GMT-06 (672 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



[Variable Thinking Wiki](#)

Life is a word problem



[Let The End Times Roll](#)



[Comenius 2008](#)

European Identity



[AIESEC-Madison](#)

Seeing the World from New Perspectives