

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](#)



Introducción

¿Qué es Ruby?

Ruby es un lenguaje multiplataforma, interpretado y orientado a objetos. Ruby fue diseñado por Yukihiro Matsumoto ('Matz') en 1993, con el Principio de la Menor Sorpresa: Matz dice "quería minimizar mi frustración mientras programo, y eso conllevaba minimizar mi esfuerzo. Este es el principal objetivo de Ruby. Quiero divertirme mientras programo. Después de lanzar Ruby y probarlo mucha gente, ellos me dijeron que sentían lo mismo que yo. Ellos fueron los que acuñaron el término de "Principio de Menor Sorpresa".

En el año 2004 hubo un boom en el interés por Ruby, debido a Ruby on Rails: el entorno para desarrollo web de David Heinemeier Hansson.

¿Cómo puede ayudarte?

En el libro de David Black "Ruby for Rails", él menciona que un buen conocimiento en Ruby puede ayudarte, como desarrollador de Rails, en cuatro formas:

1. a conocer mejor el código de tu aplicación (incluso el código que Rails escribe automáticamente).
2. a ser más productivo con Rails, incluso si dominas todas sus técnicas.
3. a familiarizarte mejor con el código de Rails, lo que te permitirá participar en discusiones sobre Rails y quizás poder ayudar detectando bugs o aportando patches.
4. a utilizar una poderosa herramienta para tareas de administración y organización conectadas con tu aplicación.

Algunas características

- **Libre de formato:** una cosa se puede hacer de distintas maneras. Escoge la que mejor se adapte a tu forma de trabajo.
- **Sensible a las mayúsculas:** dos palabras, aunque se diferencien solamente en una letra, por estar en mayúscula o minúscula, son dos cosas distintas. Por ejemplo, 'Dir' no es lo mismo que 'dir'.
- **Comentarios:** cualquier línea precedida por '#' es ignorada por el intérprete. Además, cualquier cosa que escribamos entre las líneas '=begin' y '=end', también será ignorada.

#Comentario de una sola línea

```
=begin
Esto es
un comentario
de varias
líneas
=end
```

```
=begin
Este comentario multilínea
da un error.
=end
```

MUY IMPORTANTE: este último tipo de comentarios, no puede tener espacios a su izquierda, por que daría un error. Por lo tanto, si se quiere usar, siempre van pegados al margen izquierdo de la pantalla.

- **Delimitadores de instrucción:** varias instrucciones en una misma línea pueden ser separadas por un ';', pero no son necesarios al final de una línea: este final de línea (o retorno de carro) se trata como un '\'. Si un final de línea acaba con un '\', entonces el retorno de carro es ignorado, lo que

permite tener una instrucción dividida en varias líneas.

#Varias instrucciones en una misma línea

```
a =1; b=2; c=3
#es equivalente a:
a = 1
b = 2
c = 3
```

#Una instrucción en varias líneas

```
a = 'enjuto mojamuto'
#es equivalente a:
a = "enjuto \
mojamuto"
```

- **Palabras clave:** también conocidas como palabras reservadas, son palabras en Ruby que no pueden ser usadas para otros propósitos, por ejemplo, como almacenar valores. Además, puedes estar acostumbrado a pensar que un valor 'falso' puede estar representado por 0, una cadena vacía o varias otras cosas.
- **'false' y 'nil':** En Ruby, todo esto es válido; de hecho, todo es cierto excepto las palabras reservadas 'false' y 'nil'.

page_revision: 21, last_edited: 22 Nov 2009, 15:16 GMT-06 (372 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](http://creativecommons.org/licenses/by-sa/3.0/)

Other interesting sites



[Neo Steam Wiki](#)

I put on my muumuu and pufu hat



[Life Saver](#)

Your Intelligent Life Organizer.



[DJJR Sociology](#)

a course wiki by djjr



Bisgmit