

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](#)



Alcance

El alcance es una propiedad de las variables: se refiere a su visibilidad. Los distintos tipos de variables, tienen distintas reglas de alcance. Hablemos de dos tipos de variables: las globales y las locales.

Alcance global y variables globales

Empezaremos con el alcance que menos se usa, pero no por ello menos importante: un alcance global significa que alcanza a todo el programa. Desde cualquier parte del programa, puede usarse la variable. Las variables globales son las que tienen alcance global.

Las variables globales se distinguen porque están precedidas del signo dólar (\$). Pueden ser vistas desde cualquier parte del programa, y por tanto pueden ser usadas en cualquier parte: nunca quedan fuera de alcance. Sin embargo, las variables globales son usadas muy poco por los programadores experimentados.

Variables globales por defecto

El intérprete Ruby tiene por defecto un gran número de variables globales inicializadas desde el principio. Son variables que almacenan información útil que puede ser usada en cualquier momento y parte del programa.

Por ejemplo, la variable global `$0` contiene el nombre del fichero que Ruby está ejecutando. La variable global `$:` contiene los directorios en los que Ruby busca cuando se carga un fichero que no existe en el directorio de trabajo. `$$` contiene el id (identidad) del proceso que Ruby está ejecutando. Y hay muchas más.

Alcance local

Nota: no te preocupes si no entiendes esto ahora.

Se puede intuir mirando el código dónde empieza y acaba el alcance de las variables locales, basándonos en:

- El nivel más alto (fuera de todos los bloques de definición) tienen su propio alcance.
- Cada bloque de definición de una clase o módulo tienen su propio alcance, incluso los bloques anidados.
- Toda definición de un método (def) tiene su propio alcance.

page_revision: 4, last_edited: 4 Dec 2008, 00:27 GMT-06 (726 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



Handbook

The Wikidot Handbook



Schvonn.net

~Perfection Incarnate~



The MRA Wiki

Everything in One Place



INCTR Palliative Care Handbook