

[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](#)



Azúcar Sintáctico

Algunos de los patrones de programación se repiten tanto que los lenguajes de programación incluyen formas sintácticas que son abreviaciones para estos patrones. El único objetivo de estas abreviaciones es brindar un mecanismo para escribir menos código. Es el **azúcar sintáctico**.

Por ejemplo, a la hora de cambiar un atributo:

```
class Perro
  def initialize(raza)
    @raza = raza
  end
  attr_reader :raza, :nombre    # lector
  # método modificador
  def set_nombre(nm)
    @nombre = nm
  end
end
```

```

end
end

```

```

pe = Perro.new( 'Doberman' )
pe.set_nombre( 'Benzy' )
puts pe.nombre

```

Ruby permite definir métodos que terminan en =

```

def nombre=(nm)
  @nombre = nm
end

# usando el nuevo método
nombre=( 'Benzy' )
# los paréntesis son opcionales,
# si es un sólo argumento
nombre='Benzy'

```

si empleamos "este azúcar" en el ejemplo:

```

class Perro
  def initialize(raza)
    @raza = raza
  end
  attr_reader :raza, :nombre # lector

  # modificador
  def nombre=(nb)
    @nombre = nb
  end
end

pe = Perro.new( 'Doberman' )
#pe.nombre=( 'Benzy' )
pe.nombre = 'Benzy'
puts pe.nombre

```

El signo igual es una forma familiar de asignar un valor; además que nos ahorramos poner los paréntesis.

**RAILS:** el uso del signo igual, de forma similar a la vista, es común en Rails.

page\_revision: 1, last\_edited: 4 Dec 2007, 13:46 GMT-06 (1091 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



### The SCP Foundation

To Secure, Contain, and Protect



### cheltenham tigers



## [Junko Junsui](#)



## [Micro & Epi @ Morningside Wiki](#)