

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](#)



Módulos

Los **módulos** son similares a las clases en que contienen una colección de métodos, constantes y otros módulos y definiciones. Pero a diferencia de las clases, no se pueden crear clases derivadas de los módulos.

Utilidades de los módulos

Primero, actúan como **librería**, definiendo métodos cuyos nombres no interfieran con otros definidos en otros sitios.

```
# trig.rb
module Trig
  PI = 3.1416
  # métodos
```

```

def Trig.sin(x)
  # ...
end
def Trig.cos(x)
  # ...
end
end

# moral.rb
module Moral
  MUY_MAL = 0
  MAL     = 1
  def Moral.sin(maldad)
    # ...
  end
end

# modulos.rb
require 'trig'
require 'moral'
Trig.sin(Trig::PI/4)      # "::" -> PI/4 de la clas Trig
Moral.sin(Moral::MUY_MAL)

```

Segundo, permiten aumentar las **funcionalidades** de las clases: si una clase usa un módulo, los métodos de este módulo estarán disponibles en los objetos que procedan de esta clase.

```

module D
  def initialize(nombre)
    @nombre =nombre
  end
  def to_s
    @nombre
  end
end

module Debug
  include D
  # Los métodos que actúan como preguntas,
  # se les añade una ?
  def quien_soy?
    "#{self.class.name} (\##{self.object_id}): #{self.to_s}"
  end
end

```

```

class Gramola
  # la instrucción 'include' hace referencia a un módulo.
  # Si el módulo está en otro fichero, hay que usar 'require'
  # antes de usar el 'include'.
  include Debug
  # ...
end

class OchoPistas
  include Debug
  # ...
end

gr = Gramola.new("West End Blues")
op = OchoPistas.new("Real Pillow")
puts gr.quien_soy?
puts op.quien_soy?

```

Cuando se usa `require` o `load`, se pone el nombre del item entre comillas; sin embargo, esto no hace falta con `include`. Esto es así porque en el primer caso toman strings como argumentos, mientras que `include` toma el nombre del módulo en la forma de una constante.

módulos vs clases

La principal diferencia entre heredar de una clase y usar un módulo, es que puedes usar más de un módulo al mismo tiempo. Recordar, que no se permitía tener más de una clase madre. Esto permite añadir numerosas funciones al código.

Otra ventaja, es que podemos almacenar nuestros programas de forma modular: cada módulo añade una característica. Esto también lo permitían las clases, pero la ventaja de los módulos, es que no hace falta crear una serie de jerarquías, que podría llegar a ser engorroso.

RAILS: el código de Rails hace un fuerte uso de los módulos, en particular, la técnica de reabrir las definiciones de clases y módulos.

page_revision: 1, last_edited: 13 Dec 2007, 15:17 GMT-06 (1082 days ago)

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

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

Powered by [Wikidot.com](#)

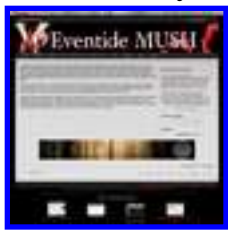
Unless otherwise stated, the content of this page is licensed under [Creative Commons Attribution-ShareAlike 3.0 License](#)

Other interesting sites



[CTwiki](#)

Il wiki di Cyberteacher



[Eventide MUSH](#)



[Evil Hat Productions Wiki](#)



[Xanadu Weyr](#)