

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](#)



Modificando clases

En Ruby, las clases nunca están cerradas: siempre se pueden añadir métodos a una clase. Esto es válido tanto para las clases que escribas, como para las que ya están incluidas con el intérprete. Todo lo que hay que hacer, es continuar con la definición de la clase:

```
require 'moto'
m = moto.new('Yamaha', 'rojo')
m.arrancar

class moto
  def informe_moto
    puts 'El color de la moto es ' + @color
    puts 'La marca de la moto es ' + @marca
  end
end

m.informe_moto
```

Ahora añadamos un método a la clase String:

```
class String
  def num_caracteres
    puts self.size
  end
end

texto = 'Cielo empedrado, suelo mojado'
texto.num_caracteres
```

Si se escribe un nuevo método que conceptualmente pertenece a la clase original, se puede reabrir el fichero de la clase, y añadir el método a la definición de la clase. Esto hay que hacerlo cuando el método es de uso frecuente, y se está seguro que no entrará en conflicto con otros métodos definidos en otras librerías que se usen más adelante.

Si el método no será usado frecuentemente, o no se quiere tomar el riesgo de modificar la clase después de su creación, crear una **subclase** (ver Herencia) es la mejor opción. Una clase puede sobrescribir los métodos de la clase de la que descende. Y es más seguro, por que la clase original permanece intacta.

Sobrecarga de métodos (methods overloading)

Las clases en Ruby sólo pueden tener un método con un nombre dado. Para tener métodos "distintos" con el mismo nombre,

se puede jugar con el número de argumentos:

```
# Un cuadrado se puede definir de dos formas:
#   Cuadrado.new([x_sup, y_izq], ancho, alto)
#   Cuadrado.new([x_sup, y_izq], [x_inf, y_der])

class Cuadrado
  def initialize(*args) # * implica número variable de argumentos
    if args.size < 2 || args.size > 3
      puts 'ERROR: Este método toma dos o tres argumentos'
    else
      if args.size == 2
        puts 'Dos argumentos'
      else
        puts 'Tres argumentos'
      end
    end
  end
end

Cuadrado.new([10,23], 4, 10)           # Tres argumentos
Cuadrado.new([10,23], [14,13])        # Dos argumentos
Cuadrado.new([10,23], [14,13],4,10)   # ERROR: Este método toma dos o tres
argumentos
```

El programa está incompleto, pero es suficiente para ver cómo se puede conseguir la **sobrecarga de métodos**.

page_revision: 6, last_edited: 26 Jul 2009, 18:04 GMT-05 (491 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



[THEKINGS](#)

A Prefect World International based faction



[Dinámica en Formación Empresarial](#)



[Portland ImmunoResearch Group](#)



[GUIDA DI STOCCOLMA](#)