

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](#)



El Primer Programa

Usemos el editor SciTE: Start/Programas/Ruby/SciTe. Se abre el editor. Para cambiar los parámetros de arranque, Options/Opens Global y allí modificar:

- `tabsize=2`, `indent.size=2` (tamaño del tabulador, y el indentado)
- `position.width=-1`, `position.height=-1` (arrancar maximizado)

una vez cambiados los valores, pulsar Ctrl+S y Ctrl+S. Para modificar estos dos valores, otra forma es:

- Ctrl+Shift+I - abre un diálogo donde modificar los valores anteriores. Tienen que ser =2.
- F11 - para maximizar minimizar la ventana.

Lo último que falta antes escribir los programas, es un abrir una ventana de terminal para ver los resultados de los programas. Para ello hay que pulsar F8.

Una vez ajustado el SciTE, lo siguiente es crear un directorio donde ir guardando los programas que vayamos creando.

'Hola'

En la ventana izquierda de SciTE (tiene que haber 2 ventanas después de pulsar F8), escribir:

```
#p001hola.rb  
puts 'Hola'
```

Ahora hay que guardar y ejecutar el programa. Las normas dicen que el nombre del fichero o directorio, es el nombre en minúsculas de la clase/módulo (más adelante se hablará de clases y módulos). Por ejemplo, la clase Foo está en el fichero foo.rb. Para guardar el fichero: File\Save As...Sálvalo como **p001hola.rb** . Todos los ficheros de código Ruby tienen la terminación ".rb". Ahora ejecuta el programa: pulsa F5. En la ventana de output aparecerá la palabra "Hola".

En Ruby, la ejecución de las instrucciones del programa, va de la primera a la última:

```
#p001hola.rb
```

No hace nada. Todas las líneas precedidas por '#' son comentarios. Y los comentarios se ignoran al ejecutarse el programa.

```
puts 'Hello'
```

puts significa put string = poner string. String es una cadena de texto. Esta instrucción saca por el output cualquier cosa que pongamos a su derecha.

Normas de código

Los paréntesis son opcionales cuando usamos un método, en este caso puts. Las siguientes instrucciones son correctas:

```
foobar  
foobar()  
foobar(a, b, c)  
foobar a, b, c
```

En Ruby, todo desde un número entero a una cadena de texto, es un objeto. Hablaremos más de esto. Y cada objeto tiene sus propios métodos (o instrucciones o funciones) que pueden ser usados para hacer cosas muy útiles. Para usar un método, necesitas poner un '.' después del nombre del objeto, y luego poner el nombre del método. Algunos métodos como puts y gets no necesitan estar asociados a un objeto, y por tanto pueden ser usados desde cualquier parte. Cuando se ejecuta una aplicación en Ruby, siempre se crea un objeto llamado main de la clase Object: este objeto es el que tiene dentro los métodos Kernel. De todo esto se hablará más adelante.

Todo esto se puede verificar por el siguiente programa (no te preocupes si no entiendes el programa, más adelante se explicará):

```
puts 'Soy una clases = ' + self.class.to_s
puts 'Soy un objeto = ' + self.to_s
print 'Los métodos del objeto son = '
puts self.private_methods.sort
```

Observar

- Programadores de C y Java - no se necesita escribir un método main.
- Los strings son secuencias de caracteres entre simple o dobles comillas. Las comillas simples son más eficientes que las dobles. Se explicará más adelante.
- Ruby es un lenguaje interpretado, entonces no hace falta recompilar para ejecutar un programa.
- Las normas de código en Ruby, establecen que el nombre del fichero/directorio tiene que ser el nombre de la clase/módulo en minúsculas, añadiendo la extensión .rb

[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



Dinámica en Formación Empresarial



Eventide MUSH



L4D Map Database

L4DMapDB.com



Pofo MultiQuiz