

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](#)



Condicionales y bucles

if,else

En Ruby, nil y false significan falso, todo lo demás (incluyendo true, 0) significan verdadero. En Ruby, nil es un objeto: por tanto, tiene sus métodos, y lo que es más, puedes añadir los métodos que se quieran.

Veamos un ejemplo de **if,else**:

```
xyz = 5
if xyz > 4
  puts 'La variable xyz es mayor que 4'
  puts 'Puedo poner más instrucciones dentro del if'
  if xyz == 5
    puts 'Se puede anidar un bloque if,else,end dentro de otro'
```

```

    else
      puts "Parte del bloque anidado"
    end
  else
    puts 'La variable xyz no es mayor que 5'
    puts 'También puedo poner múltiples sentencias'
  end
end

```

elsif

else se ejecutaba si la condición en `if` no se cumplía. Para poder tomar más decisiones, en función del valor de la variable, se usa **elsif**:

#usando `if,else` anidados

```

puts 'Hola, cuál es tu nombre?'
STDOUT.flush
nombre = gets.chomp
puts 'Hola, ' + nombre + ' .'

if nombre == 'Mojamuto'
  puts 'Pedazo de nombre!!!'
else
  if name == 'Enjuto'
    puts '...este nombre no es moco de pavo...'
  end
end
end

```

#usando `elsif`

```

puts 'Hola, cuál es tu nombre?'
STDOUT.flush
nombre = gets.chomp
puts 'Hola, ' + nombre + ' .'

if nombre == 'Mojamuto'
  puts 'Pedazo de nombre!!!'
elsif nombre == 'Enjuto'
  puts '...este nombre no es moco de pavo...'
end
end

```

#otra modificación, usando el `||` ("o" lógico)

```
puts 'Hola, cuál es tu nombre?'
STDOUT.flush
nombre = gets.chomp
puts 'Hola, ' + nombre + ' .'

if nombre = 'Mojamuto' || 'Enjuto'
  puts 'Pedazo de nombre!!!'
end
```

Además de la igualdad, existen otros operadores condicionales:

```
== igual
!= distinto
>= mayor igual que
<= menor igual que
> mayor que
< menor que
```

case

Esta instrucción es muy parecida al `if`: se crean una serie de condiciones, y se ejecuta la primera condición que se cumpla. Por ejemplo:

```
xyz = 10

if xyz % 2 == 0
  puts 'La variable xyz es par'
else
  puts 'La variable xyz es impar'
end
```

es equivalente a:

```
xyz = 10

par = case
  when xyz % 2 == 0 then true
  when xyz % 2 != 0 then false
end
```

```
puts par
```

unless

Ruby tiene una forma contraria al `if`: la instrucción **unless**. Y digo contraria, por que en `if` se ejecutaba el bloque (`do ... end`) si se cumplía la condición; con `unless` se ejecutará el bloque mientras NO se cumpla.

```
nombre = 'Pepe'
unless nombre == 'Enjuto'
  puts 'Ese nombre no tiene arte ninguno'
end
```

```
=begin
  Si el nombre no es Enjuto,
  siempre se ejecutará el bloque.
=end
```

Ejercicios

Escribe un método que pregunte por un año, y sea capaz de:

1. decir si es, o no es bisiesto.
2. poner la cantidad de minutos que tiene el año.

page_revision: 2, last_edited: 6 Aug 2010, 05:48 GMT-05 (116 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



[Trueblood: Dallas](#)



[Performance & Philosophy](#)



[Radiation Oncology - Australia](#)

Training in the RANZCR Radiation Oncology Program



Red Box Vancouver

We play old school D&D