

# Ruby Basics

Ruby: from 0 to 100 in 14400 seconds (1)

SoCraTes Canarias 2017 :-)

ruby@uniovi

Sciences

University of Oviedo

March 2015

ing language

**Welcome  
aboard!**



**github**  
SOCIAL CODING

<https://github.com/dcarral/hello-ruby>

# Setting up the environment

## 1) **Ruby Version Manager (RVM)**

<https://rvm.io/rvm/install>

## 2) **Atom (Open-source text editor)**

<https://atom.io/>

## 3) **Atom packages:**

<https://atom.io/packages/ruby-test>

<https://atom.io/packages/ruby-test-switcher>

**WAIT... WHO THE HELL**

**ARE YOU?**

# Ruby

# まつもとゆきひろ

## Yukihiko 'Matz' Matsumoto



**But... why  
Ruby?**



**Fun**

# **Principle of least astonishment**

**Poetry mode**

# Expressive

```
> 3.times { print "Ho" }
```

```
#=> "HoHoHo"
```

```
> 151.even?
```

```
#=> false
```

**Agile**

**Community**

**Matz is Nice And  
So We Are Nice**

a.k.a. MINASWAN

# Basics



A meme featuring Woody and Buzz Lightyear from the movie Toy Story. Woody is on the left, looking slightly concerned. Buzz is on the right, looking excited and holding up his right hand with three fingers extended, each with a purple ring. The background is a blurred outdoor setting.

**OBJECTS**

**OBJECTS EVERYWHERE**

> 2.next

*#=> 3 (Fixnum)*

```
class Dog  
  def woof  
    puts "Woof woof woof!"  
  end  
end
```

> Dog.class

*#=> Class*

> Dog.method(:woof).class

*#=> Method*

# Modules

```
module Animals  
  class Dog  
    # ... implementation ...  
  end  
end
```

```
module Playable
  def play
    # ... implementation ...
  end
end
```

```
class Song  
  include Playable  
end
```



# Methods

```
def factorial(n)
  return 1 if n == 0

  n * factorial(n - 1)
end
```

- Ruby methods always return the result of their last evaluated expression.
- `return` is only used to break out of the method before its end.

# Inheritance

```
module Animals
  class Dog < Animal
    # ... implementation ...
  end
end
```

- A class can only inherit from ONE other class
- There aren't Abstract classes in Ruby (in contrast to Java)

**Misc**

```
def say_hello(name)
  puts "Hello #{name}!"
end
```

```
def say_hello name
  puts("Hello #{name}!");
end
```

- In Ruby, parenthesis and semicolons are optional
- Both methods have valid syntax and are equivalent.

VAT = 0.04

total = **begin**

    sum = products.**map**(&:price).**inject**(&:+)

    sum += (sum \* VAT)

**end**

In Ruby, you can assign the result of any expression as a variable value.