

RUBY

A PROGRAMMER'S BEST FRIEND



Genís Bosch

Background

- Yukihiro Matsumoto
- Primers usos 1993
- Publicat el 1995
- Inspirat en Python i Perl
- Productivitat i diversió del programador
- Principi de la menor sorpresa

Característiques

- Llenguatge interpretat
- Multiparadigma
- Orientat a Objectes
- Altament portàtil
- Gran quantitat de llibreries estàndard

Multiparadigma

- Programació asíncrona
 - Imperatiu
 - Lògic
 - Funcional
 - Delaratiu
 - Orientat a Objectes
 - Estructurat
 - Modular
-

Més Característiques

- Nivells de variables: local, global, de classe, d'instància.
- Gestió d'excepcions
- Iteradors i clàusules
- Expressions regulars
- Sobrecàrrega d'operadors
- Recol·lecció de brossa automàtica
- Fils d'execució simultanis
- Blocs

Sistema de tipus

Fortament tipat. Sistema de tipus dinàmic.

```
x = "3"  
y = "Número " + x # "Número 3"
```

```
x = "3"  
y = x + 3 #ERROR
```

Principals aplicacions

Programes Orientats a Objectes

Mateix ús que Java, C++... Permet

Metaprogramació

“Codi per a generar codi”

Aplicacions Web

Framework Ruby on Rails

Ciència

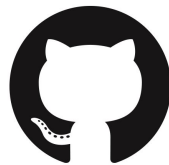
Ús científic com Medicina o Biología

Ruby on Rails

- Framework enfocat en crear pàgines web
- Combina Ruby amb HTML, CSS i JavaScript
- Aplicacions sobre servidor web
- Back-end
- Model Vista Controlador
- Convention Over Configuration i Don't Repeat Yourself
- Gemmes

Aplicaciones Web

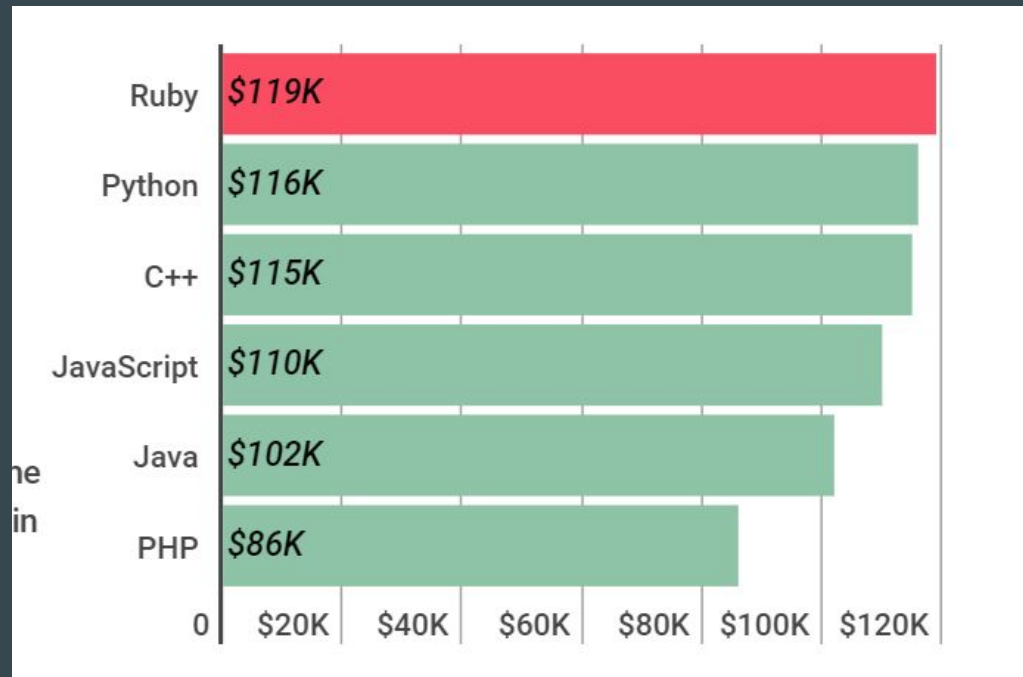
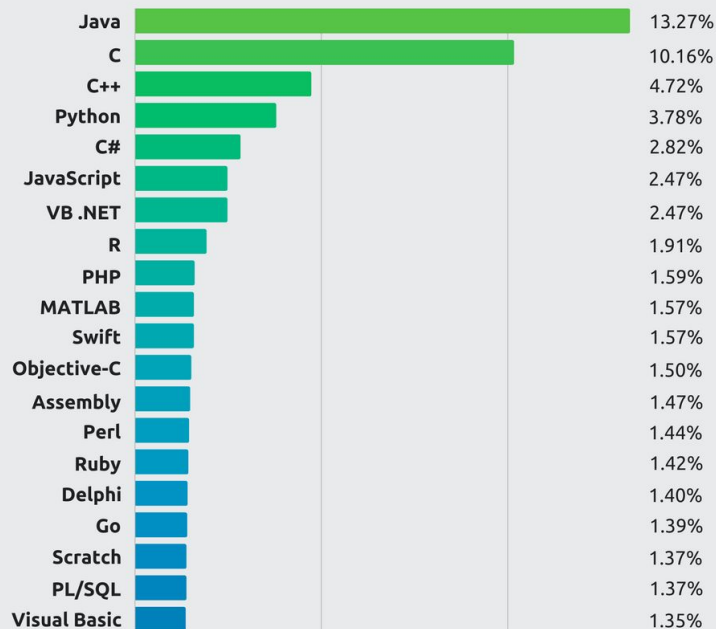
Ruby on Rails



Rànkings

Top Programming Languages

Tiobe Index - December 2017



Popularitat de Ruby

Salari a EEUU per llenguatge

Llenguatges similars

Ruby

```
def fibonacci( n )  
  
  return n if n <= 1  
  
  fibonacci( n - 1 ) + fibonacci( n - 2 )  
  
end
```

Python

```
def fibonacci(n):  
  
    if n <= 1: return n  
  
    else: return fibonacci(n-1)+fibonacci(n-2)
```

Perl

```
sub fibonacci {  
    my $n = shift;  
    $n < 3 ? 1 : fibonacci($n-1) + fibonacci($n-2)  
}
```

Exemples

```
-1.abs #1
'1234567'.length #7
1234567.to_s.length #7
3.times {print 3} #Escriu 333
rand(10).times { |x| puts x } #Conjunt de numeros aleatoris de 0 a 10
```

```
"12345".to_i #12345
"99 red balloons".to_i #99
```

```
ary = [1,2,3,4,5]
ary.each do |i|
  puts i # 1 2 3 4 5
end
```

Blocs

```
b1 = Proc.new do |param|  
  puts param  
end
```

```
b2.call
```

```
def mycompose proc1, proc2  
  Proc.new do |x|  
    proc2.call(proc1.call(x))  
  end  
end
```

```
square = Proc.new do |x|  
  x * x  
end
```

```
double = Proc.new do |x|  
  x + x  
end
```

```
double&square = mycompose double, square  
puts double&square.call(5)
```

