Ruby

EL MEJOR AMIGO DE UN DESARROLLADOR

Hash: 2934

LP - GEI FIB
PRIMAVERA 2019-2020

Propósito del lenguaje

Características:

- Dinámico
- Open Source
- Natural
- Elegante

Implicaciones:

- Simplicidad
- Productividad
- Set Up Rápido

Historia de Ruby

Nacimiento:

- Yukihiro Matsumoto
- 1993 - 1995
- Python, Perl no eran suficientes.
- Mezcla SmallTalk, Perl, Eiffel, Ada
- Productividad y diversión.

Actualidad:

- Ruby on Rails

Version	Latest teeny version	Initial release date	End of support phase	End of security maintenance phase
1.0	NA	1996-12-25[55]	NA	NA
1.8	1.8.7-p375[56]	2003-08-04[57]	2012-06[58]	2014-07-01[59]
1.9	1.9.3-p551[60]	2007-12-25[61]	2014-02-23[62]	2015-02-23[63]
2.0	2.0.0-p648[64]	2013-02-24[65]	2015-02-24[64]	2016-02-24[64]
2.1	2.1.10[66]	2013-12-25[67]	2016-03-30[68][69]	2017-03-31[70][71]
2.2	2.2.10[72]	2014-12-25[73]	2017-03-28[74]	2018-03-31[71]
2.3	2.3.8[75]	2015-12-25[76]	2018-06-20[77]	2019-03-31[77]
2.4	2.4.10[78]	2016-12-25[79]	2019-04-01[80]	2020-04-01[80]
2.5	2.5.8[81]	2017-12-25[82]	TBA	TBA
2.6	2.6.6[83]	2018-12-25[84]	TBA	TBA
2.7	2.7.1[85]	2019-12-25[86]	TBA	TBA
3.0		2020[87][88]	TBA	TBA

Legend: Old version | Older version, still maintained | **Latest version** | Future release

TABLA DE VERSIONES DE RUBY

Paradigma de programación

NATIVO:

- ORIENTACIÓN A OBJETOS
  - Métodos
  - Clases
  - Herencia/Polimorfismo

SIMULABLE:

- PROCEDURAL
  - Pseudo funciones libres
- FUNCIONAL
  - Funciones Lambda

Sistema de tipado

- Duck Typing
- Tipado Dinámico
- Tipado Fuerte

Sistema de ejecución

- Scripting

- Intérprete
  - MRI
  - CRuby
  - […]

Principales Aplicaciones

- WEB DEVELOPMENT
  - FRONT-END
  - BACK-END
- DATA ANALYSIS
- PROTOTYPING

- FRAMEWORKS
  - RUBY ON RAILS
    - AirBnb
    - Hulu
    - GitHub
    - MyFitnessPal

Características Particulares

- Expresividad
- Multiparadigma
- Problema de Escalabilidad
  - Twitter
- StartUps

- Tratamiento de excepciones
- Recolector de basura
- Multi Hilo (Independiente del OS)

EJEMPLOS DE CÓDIGO ILUSTRATIVO

VARIABLES

```ruby
# Variable Local:
## Son visibles dentro del bloque que las inicializo.
age = 10
_Age = 20

# Variable Global:
## Son visibles desde cualquier bloque.
$age = 10

# Variables de Instancia:
## Todos los objetos de una clase cuentan con la variable,
## pero el valor es privado para cada uno de ellos.
@age = 20

# Variables de Clase:
## Todos los objetos de una clase comparten esa variable
## (incluido el valor).
@@age = 20
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

IF-THEN-ELSE

```
24  a = 10;
25  b = 15;
26
27  # if statement
28  if a % 2 == 0
29      puts "Even Number"
30  end
31
32  # if-else statement
33  if b % 2 == 0
34      puts "Even Number"
35  else
36      puts "Odd Number"
37  end
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

BLOCKS

```ruby
# La funcion each permite aplicar un bloque de instrucciones
# a todos los elementos de la misma. En este fragmento podemos
# ver como llama a los elementos 'n' y los escribe.
["Esto", "son", "numeros"].each do |n|
    puts n
end
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

LOOPS

```
53  # Bucle for basico
54  for a in 1..5 do
55      puts "Salu2"
56  end
57
58  # Bucle while basico
59  i = 0
60  while i < 10
61      puts "Hey You"
62      i = i + 1
63  end
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

LOOPS

```
65  # Se puede hacer uso de breaks para controlar los bucles:
66  # Bucle do-while basico construido con breaks.
67  loop do
68      val = '7'
69
70      # using boolean expressions
71      if val == '7'
72          break
73      end
74  end
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

ARRAYS

```
150  # Los arrays pueden contener elementos de distintos tipos.
151
152  myArray = ["Como", "Me", "Gusta", "Ruby"]
153
154  myArray[1]  ## Retorna "Como"
155  myArray[-1] ## Retorna "Ruby" (los indices son ciclicos)
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

CLASES:

HERENCIAS Y POLIMORFISMO

```ruby
# Super clase Salu2
class Salu2

    def initialize

        puts "Esto es la superclase"
    end

    def super_method

        puts "Metodo de la superclase"
    end
end

# Clase Hola, subclase de Salu2
class Hola < Salu2

    def initialize

       puts "Esto es una subclase"
    end
end
```

```ruby
# Creacion de un objeto
Salu2.new

# Creacion de un objeto
sub_obj = Hola.new

# Mediante herencia, poc
# la superclase desde la
sub_obj.super_method
```

EJEMPLOS DE CÓDIGO ILUSTRATIVO

MÓDULOS

```ruby
# Creacion del modulo LP
# Utilizamos el nombre del modulo com prefijo
module LP

    Nota = 10;

    def LP.welcome
        puts "Welcome to LP!!"
    end

    def LP.tutorial
        puts "Leete el manual"
    end

    def LP.truth
        puts "Haskell > C++"
    end

end
```

```ruby
# displaying the value of
# module constant
puts LP::Nota

# calling the methods of the module
LP.welcome
LP.tutorial
LP.truth
```

Muchas Gracias