

Script Language: Ruby

ΕΠΛ 428 - Προγραμματισμός Συστημάτων
Διδάσκων: Μ.Δικαιάκος, Χ.Χρυσοστόμου

Κατερίνα Κουνούνη

Μάριος Χατζηπολλάς

Περιεχόμενα

- Ιστορική Αναδρομή
- Πλεονεκτήματα
- Μειονεκτήματα
- Ανταλλαγές
- Εγκατάσταση
- Παραδείγματα σε Ruby
- Συμπεράσματα
- Βιβλιογραφία

Ιστορική Αναδρομή

- Η Ruby είναι αντικειμενοστρεφής γλώσσα σεναρίων.
- Ο δημιουργός της είναι ο Yukihiro Matsumoto «Matz»
- Η ανάπτυξη της ξεκίνησε το Φεβρουάριο του 1993 και η πρώτη έκδοση ανακοινώθηκε το Δεκέμβρη του 1994.

Πλατφόρμες στις οποίες μπορεί να τρέξει:

- Mac OS X, είναι ήδη εγκατεστημένη.
- Στις διάφορες εκδόσεις Linux είτε είναι ήδη εγκατεστημένη είτε έχει εύκολη εγκατάσταση.
- Σε Unix, Dos, BeOS, OS/2 και Windows μπορεί να εγκατασταθεί κατεβάζοντας τον πηγαίο κώδικα και έπειτα μεταγλωττίζοντας τον.

Εναλλακτικές επιλογές με κοινά χαρακτηριστικά:

- Perl, Smalltalk, Eiffel, Ada, Lisp, and Python.

Πλεονεκτήματα

- Είναι απλή και συνεπής. Τα προγράμματα κάνουν αυτό που πρέπει να κάνουν.
- Ο προγραμματισμός είναι πιο εύκολος, πιο γρήγορος και πιο απλός από άλλες γλώσσες όπως η C.
- Μικρός σε έκταση κώδικας.
- Κάποιος με βασικές γνώσεις σε προγραμματισμό μπορεί να μάθει την Ruby πολύ γρήγορα.
- Είναι δωρεάν για να την χρησιμοποιήσεις, να την τροποποιήσεις και μπορεί να χρησιμοποιηθεί για εμπορικές εφαρμογές.

Μειονεκτήματα

- Δεν είναι πολύ διαδεδομένη και δεν έχει ευρέως χρησιμοποιηθεί στα πλαίσια της βιομηχανίας.
- Δεν υπάρχει επαρκής τεκμηρίωση σε σχέση με άλλες γλώσσες
- Έχει προβλήματα απόδοσης
- Προς το παρόν δεν μπορεί να αξιοποιήσει πολλαπλούς πυρήνες.

Ανταλλαγές - Tradeoffs

Δεν έχουμε ρητή δήλωση μεταβλητών:

- Καταλαβαίνει τον τύπο των μεταβλητών από μόνη της. Αυτό οδηγεί σε λιγότερη τεκμηρίωση όσο αφορά τις μεταβλητές, αλλά μπορεί πολύ εύκολα να οδηγήσει σε λάθη όσο αφορά τα ονόματα των μεταβλητών.

Έχουμε δυναμικούς τύπους μεταβλητών:

- Σε αντίθεση με γλώσσες με στατικές μεταβλητές χρειάζεται λιγότερη δουλειά στη δημιουργία κώδικα. Αυτό όμως οδηγεί σε σφάλματα όπως η λάθος χρήση αντικειμένων χωρίς να γίνεται αντιληπτό μέχρι την εκτέλεση, οδηγώντας στην ανάγκη για περαιτέρω έλεγχο του προγράμματος.

Εγκατάσταση εργαλείου

Βήματα για εγκατάσταση:

1. Κατεβάσετε τον πηγαίο κώδικα
 - Τελευταία έκδοση είναι Ruby 1.8.6-p111
2. Στα λειτουργικά Windows και Mac OS η εγκατάσταση μπορεί να γίνει μόνο με ένα κλικ στο ανάλογο εκτελέσιμο αρχείο.
3. Στα λειτουργικά Linux και Mac OS πρέπει ο πηγαίος κώδικας να μεταγλωττιστεί με την εντολή
 - `./install` → Linux
 - `port install ruby` → Mac OS

Το πηγαίο κώδικα και το εκτελέσιμο αρχείο μπορείτε να το προμηθευτείτε από την επίσημη ιστοσελίδα της γλώσσας Ruby:

<http://www.ruby-lang.org>

Εγκατάσταση εργαλείου

Υπάρχουν τρεις διαφορετικοί τρόποι για να τρέξεις ένα ruby πρόγραμμα

- Από αρχείο
- IRB (ruby κέλυφος με αλληλεπίδραση)
- Ως εκτελέσιμο

Για την παρουσίαση των τριών αυτών μεθόδων χρησιμοποιούμε το παράδειγμα προγράμματος που εμφανίζει το μήνυμα “Hello word”

Εγκατάσταση εργαλείου – Τρέξιμο προγράμματος από αρχείο

Βήματα που πρέπει να ακολουθήσετε:

- Δημιουργούμε ένα αρχείο, για παράδειγμα `firstExample.rb`
- Προσθέτουμε τον κώδικα που παρουσιάζουμε στην Εικόνα 1
- Έπειτα τρέχουμε το αρχείο με την εντολή `ruby`
- Το αποτέλεσμα της εκτέλεσης παρουσιάζεται στη Εικόνα 2

```
#!/usr/bin/env ruby
```

```
#this is a commend  
puts "Hello Word"
```

Εικόνα 1

```
> ruby firstExample.rb  
>Hello word
```

Εικόνα 2

Εγκατάσταση εργαλείου – Τρέξιμο προγράμματος μέσω του IRB

Με την εγκατάσταση του ruby έχουμε και το πρόγραμμα IRB. Το πρόγραμμά αυτό παίρνει την εντολή που θέλεις να εκτελεστεί και παρουσιάζει το αποτέλεσμα της εκτέλεσης στην οθόνη.

Πως το ανοίγουμε:

- Αν χρησιμοποιείς **Mac OS X** άνοιξε το τερματικό και πληκτρολόγησε `irb` και πάτησε `Enter`
- Αν χρησιμοποιείς **Linux** άνοιξε το κέλυφος και πληκτρολόγησε `irb` και πάτησε `Enter`
- Αν χρησιμοποιείς **Windows**, άνοιξε το `fxri` από το Ruby section του Start Menu.

Εκτέλεση εντολής:

Όπως φαίνεται και στην Εικόνα 2 δίνοντας την εντολή και πατώντας `Enter` το αποτέλεσμα παρουσιάζεται στην οθόνη.

```
irb(main):001:0>
```

Εικόνα 1

```
irb(main):002:0> puts "Hello World"  
Hello World  
=> nil
```

Εικόνα 2

Εγκατάσταση εργαλείου – Τρέξιμο προγράμματος ως εκτελέσιμο

Βήματα που πρέπει να ακολουθήσετε:

- Δημιουργούμε ένα αρχείο, για παράδειγμα `firstExample.rb`
- Προσθέτουμε τον κώδικα που παρουσιάζουμε στην Εικόνα 1
- Αλλάζουμε τα δικαιώματα πρόσβασης για να μπορεί να εκτελεστεί. Έπειτα το τρέχουμε ως εκτελέσιμο Εικόνα 2.

```
#!/usr/bin/env ruby  
  
#this is a commend  
put "Hello Word"
```

Εικόνα 1

```
> chmod a+x firstExample.rb  
> ./firstExample.rb  
>Hello word
```

Εικόνα 2

Παραδείγματα σε Ruby – Τύποι μεταβλητών (Local, Global)

```
$k
i0 = 1
loop {
  i1 = 2
  print defined?(i0), "\n" # true; "i0" was initialized in the ascendant block
  print defined?(i1), "\n" # true; "i1" was initialized in this block
  print defined?($k), "\n" # true; "k" was initialized in global scope
  break
}
print defined?(i0), "\n" # true; "i0" was initialized in this block
print defined?(i1), "\n" # false; "i1" was initialized in the loop
```

Ξεκινούν με	Τύπος μεταβλητής
\$	A global variable
@	An instance variable
[a-z] ή _	A local variable
[A-Z]	A constant
@@	A class variable

Παραδείγματα σε Ruby – Τύποι μεταβλητών (Class, Instance)

```
class Polygon
  @@sides = 10
  def self.sides
    @@sides
  end
end

puts Polygon.sides # => 10

class Triangle < Polygon
  @@sides = 3
end

puts Triangle.sides # => 3
puts Polygon.sides # => 3
```

```
class Polygon
  def set_foo(n)
    @foo = n
  end
end

class Triangle < Polygon
end

i = Polygon.new
j = Triangle.new
puts Polygon.foo
# => NoMethodError: undefined method foo
i.set_foo(2) #=> 2
j.set_foo(4) #=> 4
```

Παραδείγματα σε Ruby – Βρόγχοι

```
for value in 1..4
  puts "hello #{value.to_s} times"
end
```

```
10.times do
  puts "hello"
end
```

```
i = 0
while i < 10
  puts "hello"
  i += 1
end
```

```
i = 0
until i == 10
  puts "hello \n"
  i += 1
end
```

Παραδείγματα σε Ruby – Factorial

```
# Program to find the factorial of a number  
# Save this as fact.rb
```

```
def fact(n)  
  if n == 0  
    1  
  else  
    n * fact(n-1)  
  end  
end
```

```
puts fact(ARGV[0].to_i)
```

```
% ruby fact.rb 1
```

```
1
```

```
% ruby fact.rb 5
```

```
120
```

Παραδείγματα σε Ruby – Multiple Return Values

```
# if you give return multiple parameters,  
# the method returns them in an array  
# The times method of the Integer class iterate  
# s block num times,  
# passing in values from zero to num-1
```

```
def mtdarry  
  10.times do |num|  
    square = num * num  
    return num, square if num > 5  
  end  
end
```

```
# using parallel assignment to collect the return  
# value  
num, square = mtdarry  
puts num  
puts square
```

```
%ruby p019mtdarry.rb  
6  
36  
%Exit code: 0
```


Παραδείγματα σε Ruby – Κληρονομικότητα στις κλάσεις

```
class Mammal
  def breathe
    puts "inhale and exhale"
  end
end
```

```
class Cat<Mammal
  def speak
    puts "Meow"
  end
end
```

```
tama = Cat.new
tama.breathe
tama.speak
```

```
% ruby inheritance.rb
inhale and exhale
Meow
```

Παραδείγματα σε Ruby – Κανονικές Εκφράσεις

```
# p064regexp.rb
string = "My phone number is (123) 555-1234."
phone_re = /\d{3}\)\s+\d{3}-\d{4}/ #with () separate the #
                                #match string
m = phone_re.match(string)      # find in "string" the matching
                                #patterns of "phone_re"

unless m
  puts "There was no match..."
  exit
end
print "The whole string we started with: "
puts m.string                    #print all the string that
match
print "The entire part of the string that matched: "
puts m[0]
puts "The three captures: "
3.times do |index|
  puts "Capture ##{index + 1}:"
  puts m[index]
end
```

Παραδείγματα σε Ruby – Κανονικές Εκφράσεις (συνέχεια)

```
>ruby p064regexp.rb
```

```
The whole string we started with: My phone number is (123) 555-1234.
```

```
The entire part of the string that matched: (123) 555-1234
```

```
The three captures:
```

```
Capture #1: 123
```

```
Capture #2: 555
```

```
Capture #3: 1234
```

```
>Exit code: 0
```

Παραδείγματα σε Ruby – Exception Handling

```
#exceptionHandle .rb
def first_line( filename )
  begin
    file = open(filename)
    # process the file ...
  rescue
    puts "file not exist"
    filename = "STDIN"
    retry
  ensure
    puts "this is always execute"
    file.close
  end
end

first_line("my_file.txt")
```

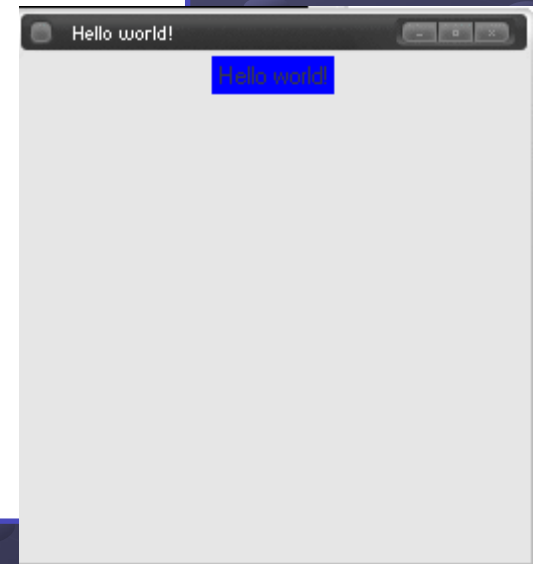
```
%ruby exceptionHandle.rb
file not exist
this is always execute
%Exit code: 0
```

Παραδείγματα σε Ruby – TCPServer, TCPSocket Classes

```
require 'socket'
server = TCPServer.new("127.0.0.1", 80)
loop do
  socket = server.accept
  while socket.gets.chomp.length > 0
  end
  socket.puts "HTTP/1.1 200 OK"
  socket.puts "Content-type: text/html"
  socket.puts "" socket.puts "<html>"
  socket.puts "<body>"
  socket.puts "<div>"
  socket.puts "<b>Yay!Your first networking application!</b>"
  socket.puts "</div>"
  socket.puts "</body>"
  socket.puts "</html>"
  socket.close
end
```

Παραδείγματα σε Ruby – GUI χρησιμοποιώντας TCL/TK

```
require 'tk'  
root = TkRoot.new  
do  
    title "Hello world!"  
    minsize(250,250)  
  
end  
TkLabel.new(root)  
do  
    text 'Hello world!'  
    background 'blue'  
    pack { padx 15; pady 15; side 'left'  
  
end  
Tk.mainloop
```



Παραδείγματα σε Ruby – Αλλαγή ονόματος αρχείων

```
require 'fileutils'
include FileUtils

if ARGV.length != 1
  puts "usage: prepend \"string to add to filenames\"\n"
  exit
end

strPrepend = ARGV[0]

#get list of files
fileList = Dir.entries(".")
```

Παραδείγματα σε Ruby – Αλλαγή ονόματος αρχείων (συνέχεια)

```
#adds string to beginning of file names
begin
  for i in fileList
    if(File.directory?(i) == false)
      newName = strPrepend + i
      if(i != newName and File.exists?(newName) == false)
        File.rename(i, newName)
      end
    end
  end
end

#If an error occurs, print a notice
rescue Exception
  $stderr.print "\nAn Error has occured while processing file " + i
  + "\n\n"
  $stderr.print "Exiting Program \n"
end
```


Παραδείγματα σε Ruby – Ταξινόμηση αρχείων

```
require 'optparse'
require 'fileutils'
include FileUtils

fileList = Dir.entries(".");
letters= 'A'..'Z'

begin
  for i in letters
    if(File.directory?(i) == false)
      Dir.mkdir(i)
    end
  end
end

if(File.directory?("0-9") == false)
  Dir.mkdir("0-9")
end
```

Παραδείγματα σε Ruby – Ταξινόμηση αρχείων (συνέχεια)

```
#Exit if there is an error
rescue Exception
  $stderr.write "\n An Error occured while creating directories
\n\nExiting Program\n"
  exit
end
```

```
#sorts the files into directories
begin
  for j in letters
    for i in fileList
      if(File.directory?(i) == false)
        #Sort files that start with a letter
        if(i.upcase.index(j) == 0)
          if(File.exist?(File.join(j, i)) == false)
            cp(i, File.join(j, i))
            rm(i)
          end
        end
      end
    end
  end
end
```

Παραδείγματα σε Ruby – Ταξινόμηση αρχείων (συνέχεια)

```
#sort files that start with a digit
  if(i.index(/[0123456789]/) == 0)
    if(File.exist?(File.join("0-9", i)) == false)
      cp(i, File.join("0-9", i))
      rm(i)
    end
  end
end
end
end
end

#Exit program if there is an error
rescue Exception
  $stderr.print "\nAn Error has occured while processing file " + i + "\n\n"
  $stderr.print "Exiting Program \n"
  exit

end
end
```

Συμπεράσματα

- Η Ruby έχει απλή σύνταξη.
- Γνωρίζοντας βασικό προγραμματισμό μπορείς εύκολα να την χρησιμοποιήσεις.
- Συνδυάζει script language και object oriented. Με το συνδυασμό αυτό παράγουμε μικρό σε μέγεθος κώδικα και εύκολα συντηρήσιμο.
- Παρέχει ένα μεγάλο αριθμό από μεθόδους για κάθε αντικείμενο καθιστώντας πιο εύκολη την δουλειά του προγραμματιστή.

Βιβλιογραφία

- Ruby home page: <http://www.ruby-lang.org/en/>
- A Ruby introduction was presented to the New York City CTO Club: <http://www.io.com/~jimm/downloads/rubytalk>
- CMSC 330: Organization of Programming Languages “Introduction to Ruby” <http://www.cs.umd.edu/class/spring2008/cmsc330/lectures/02-ruby.pdf>
- Meshplex the tutorial database http://www.meshplex.org/wiki/Ruby/Ruby_on_Rails_programming_tutorials
- I.M. Testy: Disadvantages of Ruby for Test Automation <http://blogs.msdn.com/imtesty/archive/2006/06/08/621755.aspx>
- Wikipedia [http://en.wikipedia.org/wiki/Ruby_\(programming_language\)](http://en.wikipedia.org/wiki/Ruby_(programming_language))

Τέλος Παρουσίασης

ΕΡΩΤΗΣΕΙΣ;