CSCI 3055U Course Project – Ruby

Alexander Marshall - 100487187

Language Design

- Type System
  - Ruby is dynamically typed so it is not necesarry to specify types for any variables.

- Loop Syntax
  - To do a simple action multiple times, you can do:
    'n.times { <action here> }', where 'n' is an integer.

  - The while-loop while has the syntax 'while <condition>'. Parenthesies are not needed for loop conditions.
  - For loops are done in the same way as while loops and you can also do the following:
    'for i in 1..10' or 'for i in 1...10'.

    - The first one goes from 1 to 10 *inclusively*, while the second one goes from 1 to 10 *exclusively*.

  - Ruby also has syntax for exiting loops like 'unless' which functions similarly to an if-statement.
    - There is also 'break if' which is an if-statement that automatically breaks the loop if the specified condition is true.
  - All loops and conditional statemends must be closed with the 'end' keyword.
  - You can iterate over arrays using the '.each' syntax

- Function Syntax

- To start a method: 'def method_name (var1, var2)' and to end a method, use the 'end' keyword.
  - Input parameters can also have default values. This is done in the following way:
    'def method_name(var1 = val1, var2 = val2)'

- To call this method with two parameters, we do not need to use parenthesies, but we do need commas. 'method_name 25, 30'

- Functions as Values
  - The closest thing to functions as values in Ruby is through the Proc class.
  - You can define a method that returns an instance of Proc, and then 'call' that procedure. See below:

```ruby
def gen_times(factor)
  return Proc.new {|n| n*factor }
end

times3 = gen_times(3)
times5 = gen_times(5)

times3.call(12)               #=> 36
times5.call(5)                #=> 25
times3.call(times5.call(4))   #=> 60
```

    - source: http://ruby-doc.org/core-2.2.0/Proc.html
  - I haven't played around with this at all, but it seems like a useful tool.

- Closure
  - Blocks are the most commonly used form of closures in Ruby.
    - A block is simply a 'chunk' of code.
    - Each call to yield in the method will invoke the block

that was passed to that method.
    - You can specify arguments to yield what will be passed to the block.
    - Each method may only receive a single block argument but it may yield to this single block multiple times.

  - A Proc (procedure) is a closure and an object that can be bound to a variable and reused.
    - A Proc is declared by using proc and then defining a block of code.
    - A Proc can be invoked using the 'call' method.

  - Lambdas are a more strict form of Proc.
    - They are defined with ->(<code here>)
    - They are more strict with argument checking and can return a value with the 'return' keyword.

  - Finally, there is the '&' operator.
    - It works differently depending on what it is applied to.
    - Placing it in front of a Proc will convert it to a block and vice versa.
    - Placing it in front of anything else will call .to_proc on the object and then convert it to a block.

- Anonymous and High Order Functions
  - Anonymous methods in Ruby can be defined in the following ways:

```ruby
proc {|arg| puts arg}
Proc.new {|arg| puts arg}
lambda {|arg| puts arg}
->(arg) {puts arg}          # introduced in Ruby 1.9
```

    - source: http://en.wikipedia.org/wiki/Ruby_

%28programming_language%29

Sample Program

See 'hangman.rb', included with this submission. The code may not be elegant, but that's mostly because I haven't spent an extensive amount of time with Ruby. Ensure you have Ruby installed, then run the program with 'ruby hangman.rb'. Have fun!

Discussion

- Ruby Standard Library
  - The Ruby standard library seems to be fairly expansive, as I did not need any additional libraries to complete my program.
  - Additional libraries are released in the form of a 'gem'.
  - RubyGems is a Ruby packaging system that is designed to facilitate the creation, sharing, and installation of libraries.
  - You can find gems here: https://rubygems.org/
- Development Tools
  - Some Ruby IDEs include Aptana, RubyMine, Redcar, TextMate, and Netbeans.
    - All of these have support for Ruby on Rails.
- Ruby Forums and Activities
  - Ruby has forums at www.ruby-forum.com and Ruby on Rails specific forums at www.railsforum.com.
  - You can learn Ruby at www.codeacademy.com, which is how I learned it.
  - I had no trouble solving my problems through previous questions asked on www.stackoverflow.com.
- Popular Open Source Ruby Projects

- GitLab
  - GitLab is a software that allows users to work together on code through git repository management.
  - It runs on Ruby on Rails.
  - https://about.gitlab.com/

- Errbit
  - Errbit is an error catcher designed for special purposes and special users.
  - Errbit also uses Ruby on Rails.
  - http://errbit.github.io/errbit/

- Kandan
  - Kandan is a web based team chat which includes features like searchable message history, file uploads, and shared room audio.
  - Kanada uses Ruby on Rails.
  - https://github.com/kandanapp/kandan

- Check out http://www.opensourcerails.com/ to see more cool projects that use Ruby on Rails!

- Uses of Ruby in Practice
  - Our beloved GitHub uses Ruby on Rails.
  - Twitter is implemented using Ruby and also uses Java, Scala, and JavaScript
  - Metasploit is the worlds largest Ruby project (according to Wikipedia), with over 700,000 lines of code.
    - It is the “world's most used penetration testing software.”
    - Learn more here: https://www.metasploit.com/