Ruby on Rails Course -1

What Ruby used for?

Some **useful links** that come handy when you are going through the course material:

Ruby Download
- http://www.ruby-lang.org/en/downloads/
Ruby Online
- http://tryruby.hobix.com/
Duck Typing
- http://en.wikipedia.org/wiki/Duck_typing
- http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/100511

Jargon to watch out:
Duck Typing, Operators, Methods, Smalltalk

Download the detailed Ruby on Rails Course contents.
Reach me @ tosumanthkrishnaATgmailDOTcom for more discussions.

Duck typing:

```
Duck typing is way of thinking about programming in Ruby.

For folks who come from languages such as Java or C++, types are
basically the same as classes. When you ask 'what is the type of
"cat"?', the answer comes back as 'String'.

These languages use the type==class model as the model for programming.
You say

     String fred;

to say that 'fred' has type 'String', and the language implements that
by saying that fred can only reference objects that are class String.

Ruby doesn't use that model. In Ruby, types are defined as the
capabilities of objects. Classes can be used to give objects their
initial capabilities, but from that point on, the class is (almost)
irrelevant. When I write

     fred << "dog"

in Ruby, I don't care whether fred is a String, a File, or an Array (a
fact that's very useful when writing unit tests).
```

What ruby is used for?

Let us recollect the statement:
Ruby is "an interpreted scripting language for quick and easy object-oriented programming".
Ruby is used for many kind of applications be it writing automation scripts and run on your local pc's, or developing automation tools, or Website, or Blogs, Web Application or web based products and the list goes on...

We here pick up basics of Ruby just enough to get going with Rails and start developing simple websites and web applications.
All the things that we had said about ruby in earlier post What is Ruby? will be discussed in coming posts.
Before even going for installing the ruby in your desktop, let us experiment on ruby with small examples and work/execute them here online.

Learning through Examples:

This is our play ground.
This is called interactive ruby shell and we find ">>" similar to command/shell prompt. Now let's start with famous example of printing out "Hello World!" statement.

1. type puts "Hello World!"

   >> puts "Hello World!"
2. then click "Enter" button
3. Now you would find the expected returned

- >> puts "Hello World!"
- Hello World!

  => nil

So puts is the ruby way of printing the things to console/prompt. The first line means to print the string within the quotes(i.e. Hello World!). Your would also find "=> nil" displayed in the next line of the Hello World! statement, `puts` always returns nil, which is Ruby's absolutely-positively-nothing value.

Going on to the next example:

- Ruby has simple syntax.
- Ruby is dynamically typed.
- Ruby needs no variable declarations.
- Ruby operators are also methods.

Now I want to do simple math. Based on the above points,
to add two integers just type : 4+2
to subtract two integers just type : 4-2
to multiply two integers just type : 4*2
to divide two integers just type : 4/2
You can check out with other types also. Need not be of same type, since ruby is dynamic language at the run time they would be evaluated.

for example you type 2.0+3 and see for yourself.

Coming on to the variable declaration, ruby does not require the variable declarations. Let us the see this point also:
you type
>> a=5
=>5
>>b=2.0
=>2.0
>>c=a+b
=>7.0
As you see we have not declared the variables a,b,c and what type they hold. All these variables are evaluated at the run time and treated accordingly.
Ruby treat the operators(+,-,*,/,%...) as just like methods, which is conceptually derived from Smalltalk.

Examples:

>> puts "Hello World!"
Hello World!
=> nil
Ruby is dynamically typed.
Ruby operators are also methods.

#Addition
>> 3+2
=> 5
#Subtraction
>> 5-1
=> 4
#Multiplication
>> 5*2
=> 10
#Division
>> 10/2
=> 5

#Modulus Division
>> 10%2
=> 0
Ruby needs no variable declarations.
Variable declaration. No type is mentioned.
>> i=5
=> 5

```
>> k=9
=> 9
#   Adding two variables
>> i+k
=> 14
#   Adding two variables and assigned to another variable
>> c=i+k
=> 14
>> c
=> 14
#   Float assigned to “a”
>> a=5.0
=> 5.0
#   Integer assigned to “b”
>> b=2
=> 2
# The result of adding them is float.
>> a+b
=> 7.0
```

Discuss/Comment @ TechSavvy

Download Ruby Resources

- RoR Course Structure
- What is Ruby?

View Ruby Resources @ Scribd

- RoR Course Structure
- What is Ruby?