Day of Ruby - Pre-Arrival Instructions

Congratulations on choosing Ruby in your future, this is the beginning of great things.

On Saturday you will arrive at the Miami University Voice of America Learning Center with your favorite computer ready and setup for the day. If you have any problems with the setup yourself then come early and we will help you.

> http://muohio.edu/voalc

There will be water and coffee available when you arrive and lunch is provided.

NOTICE: The start time has changed. Due some room scheduling issues the actual start has been pushed back half an hour to 9:30.

Registration	Foyer	8:00-9:30
Laptop setup help	Room 111	7:30-9:30
Introduction to Ruby with Koans (EdgeCase's Jim Weirich)		
	Auditorium	9:30-12:00
Lunch (provided)		12:30-13:30
Rails in a day (Gaslight Software's Chris Nelson & Doug Alcorn)		
	Auditorium	13:30-16:30

Setup Instructions

There are two sets of instructions below, one for the morning session with Jim Weirich and the second for the afternoon session with Chris Nelson and Doug Alcorn.

Both sessions will use the same Ruby environment so this is what you need first.

Ruby Setup Instructions

We will be using version **1.9.2-p0** for the Rails Intro as it it now the most recent official release. There's no need to have any other version if this is your first and only use of Ruby, but if you already have ruby and need to keep it around (and if you have Mac OS X, you do!), then look at `rvm` (or `pik`) first. To see if you already have ruby, run this command:

    ruby --version

Windows:

The simplest way to get Ruby running on your windows machine is with the One Click Installer. You want to pick the 1.9.2-p0 version:

> http://rubyinstaller.org/downloads

Mac OS X:

There has been a ruby interpreter in OS X since 10.5. You certainly have an older version as even Snow Leopard (10.6) only has 1.8.7. You want to install rvm (see below) and then run:

```
rvm install ruby-1.9.2-p0
rvm use ruby-1.9.2-p0
```

Git

The source management tool of choice is `git`. It is fully distributed meaning that once you start working with a repository, you have a **full** copy of everything locally (and so does everyone else). Since we'll be using git to obtain some of the latest version of other tools, it's worth getting it installed first.

http://git-scm.com/download

Depending on your platform, there are other tools that might make working with git nicer:

http://git-scm.com/tools

rvm (Unix-like systems including Mac OS X)

Since Ruby-1.9.2-p0 is so new, it's unlikely to be the standard, preinstalled ruby on your system. The easiest way to manage multiple ruby versions in with a took called rvm.

http://rvm.beginrescueend.com/
http://rvm.beginrescueend.com/rvm/install/

tl;dr? -- just run this:

```
bash < <(curl http://rvm.beginrescueend.com/releases/rvm-install-head)
```

Then go back and follow the "Post Install" instructions on
http://rvm.beginrescueend.com/rvm/install/

pik (Windows systems)

If you already have an earlier ruby, presumably you have an environment for compiling and this pointer to pik is all you need to get going:
http://github.com/vertiginous/pik#readme

Getting the initial code…

Ruby Koans

Visit the web site where there are detailed instructions to guide you through the setup. Note that the koans work for any ruby from 1.8 onward, but we'll be using 1.9 (specifically, 1.9.2-p0).
http://rubykoans.com/

Don't forget to follow the "Download the Koans" link so that you're ready to begin the *Path to Enlightenment.*

You'll need some kind of editor and the Ruby Koans page has a good list.
> http://rubykoans.com/#what_editors

If you don't already have a text editor with which you are familiar, we'd suggest trying RubyMine. If you're going to try RubyMine, be sure to get the version that is "Rails 3 ready" by looking for the "Next Release" link near the top of the download page.

Rails in a Day

Rails itself and much of distributed ruby code comes packaged in the form of "gems" or "rubygems." Ruby 1.9 has built-in support for rubygems so there's nothing specifically to install to start using gems.

To deal with version dependencies, there is a gem called bundler that we will install first.
```
gem install bundler
```
Then we will get the initial project repository:
```
git clone git://github.com/Cincinnati-RB/Suggestion-Box.git
```

If you are on Windows using the Git GUI and need to specify the source:
```
Source: git://github.com/Cincinnati-RB/Suggestion-Box.git
Target: C:\Repositories\Suggestion-Box
```

You can specify any location that's convenient in place of `C:\Repositories\`, but Git GUI will want to create the `Suggestion-Box` directory for you so it must not exist.

Change to the new Suggestion-Box directory and get everything else installed.
```
cd Suggestion-Box
```

The branch that holds all the code that we will be using is called "day_of_ruby" so let's make sure that exists locally:
```
git checkout day_of_ruby
```

We will be starting at the point in the repository tagged with `dor_start` and you'll be doing your own work in a local branch that we'll call "`private`" that you should create now:
```
git checkout -b private dor_start
```

Now we can get bundler to install the other gems that we'll be using:
```
bundle install
```
If you get a strange error containing something like "[] not defined for NilClass", delete the `Gemfile.lock` file and try again.

Mac OS X 10.5 (Leopard) users may (i.e., probably will) get an error during the install of the sqlite3-

ruby gem. To fix this, you'll have to change to an earlier version of the gem by editing the `Gemfile` to add the "1.2.5" version requirement to sqlite3-ruby (line 4):

```
gem "sqlite3-ruby", "1.2.5", :require => "sqlite3"
```

Then rerun the `bundle install` and it will say it is "Using..." all the other gems it already installed, install the sqlite3-ruby-1.2.5, and replace the vendor/cache/sqlite3-ruby-1.3.1.gem file with a new vendor/cache/sqlite3-ruby-1.2.5.gem. Since you've already changed to a local, private branch (that we so conveniently named "`private`"), this change won't affect anyone else.

Windows users may get an error during the install of the `escape_utils` gem. We don't really need this. We can fix it by editing the `Gemfile` to remove the requirement for `escape_utils` (line 6):

```
gem "escape_utils"
```

Then rerun the `bundle install` and it will say it is "Using..." all the other gems it already installed. Since you've already changed to a local, private branch (that we so conveniently named "`private`"), this change won't affect anyone else. There's also an initializer to remove if you've had to remove the gem.

```
git rm config/initializers/escape_utils.rb
```

Mac OS X users will already have sqlite3 installed, but Windows (or Ubuntu, etc.) users may not. The test is to run the commands to create and initialize the database. (Anybody could run these and we'll be doing this as we get started Saturday if you haven't done so already.)

```
rake db:create
rake db:migrate
```

If the db:create fails, you don't have the library for sqlite3. Windows users should download:

http://www.sqlite.org/sqlitedll-3_7_2.zip

and put the sqlite3.dll into a location in the load path (e.g., `C:\Windows\System32\`).

Linux users should use their package management system to do the install. For example, in Ubuntu or Debian:

```
apt-get install sqlite3
```

If you have changed the files locally, you may have some cleanup to do so that your working area for git is "clean." If you are familiar with git, you can make a commit to your local branch `private`, or just remember to get some help during the setup time.