

Class 30: Ruby (3)

Held: Friday, April 13, 2007

Summary: Today we conclude our explorations of Ruby by considering a number of topics.

Related Pages:

- EBoard.
- Reading: Walton & Hibbs - Rolling with Ruby on Rails Revisited and Walton - Rolling with Ruby on Rails Revisited, Part 2.

Notes:

- Beware! Friday the 13th falls on Friday this month.
- Don't forget next week's Tuesday extra.
- I'd like to hear prospective groups on Monday and prospective topics on Friday.

Overview:

- About the Presentations.
- Rails.
- Convention and Introspection.
- Ruby vs. Java, Revisited.

Presentations

- The last two weeks of the semester are reserved for student presentations.
- You should form groups of three to give these presentations.
- You may choose almost any topic related to programming languages, but you must clear it with me first.
- You should plan a thirty-minute presentation.
 - If you want to use a different structure (e.g., a lab), please clear it with me first.
 - You should rehearse your presentation in advance.
- I expect you to support both auditory and visual learners
 - You should probably say stuff, but you also need to write stuff.
 - Powerpoint is fine.
 - Using the whiteboard is fine.
- You will be responsible for assigning a reading in advance.
- You should choose a topic by next Friday.
- ...

Ruby on Rails

- I understand that much of Ruby's current popularity has more to do with Rails than any other aspect of Ruby.
 - Rails designers would probably tell you that Rails also benefits closely from Ruby's design.
- After the readings, you should have a partial sense of the applications and structure of Ruby.
- No, I don't really believe the 10x increase in productivity figure, but I've also read a few articles by former nonbelievers who have tried Rails and found that they did, in fact, have a similar increase.
- So, what parts of Web application development does Rails automate?
- What aspects of Ruby are likely to help with this? (Or is that Rails just happens to be written in Ruby?)

Convention and Introspection

- Although our readings do not say so explicitly, I expect that the emphasis on convention in Rails relates closely to Ruby's introspective capabilities.
- The typical context for this is unit testing.
- Most unit testing frameworks do something like the following:
 - Look for all the Unit tests (whose name or location is determined by convention or configuration); each of which is a class.
 - For each class, look for all the methods intended to do testing. Convention often specifies a prefix for those methods (e.g., they start with the word "test").
 - For each test method, call it, accumulating results.
 - The test methods use library methods for the individual tests.
 - The library methods accumulate results.
- In

Ruby vs. Java, Revisited

- What do you see as the key differences between Ruby and Java?
- What do they suggest about the intended perspective, audience, purpose, or whatever of each language?