Ruby, Part 1 of 6: Introduction and Classes

page 1

Meet the expert: Jordan Hudgens has certifications for Ruby on Rails, Thinkful; Ruby on Rails, Bloc.io; Front End Development, Thinkful; and AngularJS, Thinkful. He is currently vice president of engineering for TRACKR in Midland, Texas and is working on his PhD in Computer Science from Texas Tech. In addition to Ruby, Jordan works with PHP, JavaScript, MySQL, Postgres, CSS3, C, C++, C#, Objective-C, and Python. He also works with the frameworks Rails (Ruby), Zend (PHP), and Django (Python), plus the libraries AngularJS, jQuery, and Backbone.js.

Prerequisites: You should be familiar with basic text editing in a Windows or Linux environment. No prior knowledge of programming is assumed.

Runtime: 01:59:41

Course description: Join programming expert Jordan Hudgens for an introduction to the Ruby programming language. In this course, he will demonstrate what makes Ruby unique as a language and how it can be leveraged to build robust applications. Hudgens will also provide a step-by-step guide on how to install Ruby on various operating systems as well teach you best practices to ensure that you are building programs that fall within guidelines of the wider Ruby development community.

Course outline:

History and Installation
- Introduction
- History of Ruby
- Tools We'll Use
- Demo: Sublime Text
- Demo: IRB
- Installing Ruby on a PC
- Demo: RubyInstaller for PC
- Installing Ruby on a Mac
- Demo: Ruby on Mac
- Ruby on a Mac + RVM
- Demo: RVM
- Summary

Conventions and Best Practices
- Introduction
- Common Ruby Conventions
- Demo: Ruby Styles
- Ruby Best Practices
- Demo: Best Practices
- Summary

Classes
- Introduction
- Ruby Classes
- Demo: Creating a Class
- Class Accessor Methods
- Demo: Accessor Methods
- Inheritance
- Demo: Inheritance
- Demo: Inheritance Cont.
- Summary

Methods
- Introduction
- Methods
- Demo: Passing Arguments
- Demo: Optional and Splat
- Class vs. Instance Methods
- Demo: Class vs. Instance
- Summary

Object Oriented Programming
- Introduction
- Object Oriented Nature
- Demo: Check for Objects
- Method Lookup
- Polymorphism
- Demo: Polymorphism
- Summary