Ruby, Part 6 of 6: Error Handling, Frameworks, and Algorithms

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.
This course builds on the previous ruby courses:
Ruby: Introduction and Classes, Ruby: Variables and Data Structures, Ruby: Input/Output and Gems.

Runtime: 01:43:19

Course description: Ruby expert Jordan Hudgens covers error handling, web frameworks, and algorithms in Ruby in this course. He will begin with a walk through the syntax for basic error handing, emphasizing best practices, followed by a review of the two most popular web frameworks for Ruby: Sinatra and Rails. Hudgens will show how to build two web applications, with the Rails app being a functional blog that includes full CRUD features, and then walk through implementing popular algorithms in Ruby and building the insertion sort and quicksort algorithms. Finally, he will discuss creating a binary search program.

Course outline:

Error Handling & Syntax
• Introduction
• What Is Error Handling?
• Begin/Rescue
• Demo: Error Handling
• Summary

Antipatterns
• Introduction
• Antipatterns
• Demo: Antipatterns
• Summary

Error Logger
• Introduction
• Error Logger
• Demo: Error Logger
• Summary

Sinatra
• Introduction
• Sinatra
• Demo: Sinatra
• Demo: Parameters
• Summary

Rails
• Introduction
• Ruby on Rails
• Demo: Getting Started
• Demo: Migration
• Demo: Preview Blog
• Summary

Insertion Sort
• Introduction
• Why Learn Algorithms?
• Algorithm Development
• Demo: Insertion Sort
• Demo: While Loop
• Demo: Test Insertion Sort
• Summary

Quicksort
• Introduction
• Sorting Algorithms - Quicksort
• Demo: Quicksort
• Demo: Test Quicksort
• Summary

Binary Search
• Introduction
• Binary Search
• Demo: Binary Search
• Demo: Test Binary Search
• Summary