



The Learn-Verified Full Stack Web Development Program

Overview

This online program will prepare you for a career in web development by providing you with the baseline skills and experience necessary to obtain an entry-level software development job and be immediately productive in such a role. While the curriculum is designed to teach you to think—and build—like a software engineer (independent of any specific language) the bulk of the material covered is based on the Ruby ecosystem.

You will develop a foundation in programming fundamentals, and conquer the concepts of object-oriented programming. You will work with APIs (Application Programming Interfaces), become proficient in database modeling and ORM (Object Relational Mapping), understand the concept of MVC the (Model View Controller) Framework, and execute application deployment. Labs are taught using test-driven development, allowing you to gain a real-world programming experience, while giving instructors on the Learn team the ability to evaluate and assist in real time.

Throughout the program, you will be challenged to solve hundreds of test-driven labs, develop several self-directed projects, and undertake live assessments with Flatiron School instructors to verify your progress. By the end of the program, you will have built fully functional web-based applications. You will be encouraged to expand your digital portfolio by maintaining active technical blogs and getting out into your local development community to attend and give talks at public meetups. The program culminates in a final verification with a Flatiron School instructor. Upon satisfactorily completing that verification, you will move into the job placement process with the aim of helping you land a job offer within six months, or your money back.

Why Ruby?

We will leverage the Ruby programming language for three primary reasons:

Readability

Much of the initial difficulty in learning programming stems from the learning curve necessary to gain comfort with a language's syntax. While traditional languages like C++, Java, and even Python employ white-space sensitivity and heavy use of constructs like brackets and semi-colons, Ruby is designed to be more readable and accessible, allowing new programmers to focus immediately on the fundamental concepts and logic, rather than basic syntax.

Open Source

The Ruby language has nurtured an incredible open source community. This will allow you to leverage free, publicly available tools to build applications with complexity and real-world application beyond what you would ever approach otherwise.

Career Flexibility

The Ruby language allows students to explore abstract programming frameworks via a low-level, object-oriented language. As such, you are able to gain immediately relevant career skills, while gaining a foundation in a language that will afford you the ability to easily expand on your knowledge base into other relevant skillsets. Many students leverage the foundation they learned in Ruby to accept full-time positions that leverage other languages ranging from Python and JavaScript to Objective-C.

Using Learn

Real Tools

You can't learn real skills without real tools. We don't believe in contrived environments or multiple-choice quizzes. Learn users set up a real development environment and work through test-driven labs with a git-based workflow.

Real Time Chat

When you're stuck, you'll get help. Right away. No need to wait a week to meet with your mentor. No emails to send. No forums to post to. Be part of a community of active Learners and experts with whom you can work to solve problems and build software.

Learn at Your Own Pace

Move as fast as you'd like. We're here supporting you, making sure you're hitting your goals. You're not held back by an artificial schedule. Whenever you're learning, whatever your progress, you can expect other Learners to be online with you, learning together.

Open Curriculum

The curriculum on Learn is entirely open-sourced on GitHub. It's been consumed by hundreds of students across dozens of classes and is continually improved with feedback from students, instructors, employers, and the community at large.

Transparent Work History via GitHub

All of your work—by which we mean every lesson you've ever completed on Learn—ends up on GitHub. No badges. No certificates. No degrees. Instead, a fully transparent record of your work in a way that actually matters to employers.

Curriculum Overview

1

The “Intro to Ruby Web Development” Track

Before starting the Learn-Verified program, you will be required to complete a ~30 hour track on Learn that will take you through the fundamentals of Ruby programming including variable, methods, control flow, looping, arrays, and user input. You will learn the mechanics of test-driven development, will build a command-line application, and be introduced to the basics of object-orientation.

2

The Learn-Verified Full Stack Web Development Track

The Learn-Verified Full Stack Web Development Track involves the following:

- 600-800 hours of content covering the entire web development stack
- 100+ readings, videos & interactive quizzes
- 200+ test-driven labs
- 3 peer-reviewed projects
- 2+ live assessments with an instructor to check your progress
- 1 final evaluation with an instructor to obtain your Learn Verification

The Full Stack Web Development curriculum covers eight high-level topics:

Topic: Ruby

You will begin exploring version control using git commands including with cloning, branching, merging, rolling back commits, forking, and submitting pull requests.

You will then learn fundamental concepts in programming including repls, methods, loops, variables, variable scope, conditionals, blocks and iterators, case statements, arrays, scope, hashes, regular expressions, iterators, enumerables, data structures, nesting, etc.... Topics build in complexity and provide the foundation for the rest of the course.

We will help you embrace error messages as clues and gain a fundamental appreciation for failure as the only way to learn and progress. You will gain experience in debugging with various gems and tools designed to track down issues in code.

Curriculum Overview

(cont'd)

Topic: Object Orientation

You will gain experience with Object-Oriented Programming and understand how it allows programmers to bundle code and create reusable objects and methods, allowing for increasing complexity in software.

Topic: Object Relational Mapping & SQL

ORM (Object Relational Mapping) allows programmers to query and manipulate data from a database using an object-paradigm. You will learn to write and manipulate data using the Ruby language. You will gain an appreciation for the structure of a database, how to map out tables, and the difference between the various table relationships. You will learn how to wireframe database structures, as well as how to link your applications to a database. You will also cover SQL, domain modeling, relational database theory, schema architecture, and the Object Relational Model, including the ActiveRecord pattern.

Topic: Rack

This unit is designed to give you an understanding of HTTP and how the Internet works, as implemented through the Ruby web interface of Rack. You will build your own HTTP servers and learn how the request / response model of the web works. Your servers will listen to HTTP requests and respond with well-formed HTML responses. You will learn to understand the web with the few abstractions provided by the tool set.

Topic: Sinatra

Sinatra is a Domain Specific Language (DSL) written in Ruby for building web applications on top of Rack. This framework provides you with exposure to design patterns in web applications. The topics covered in this unit include architectural patterns such as REST (Representational State Transfer), MVC (Model-View- Controller), HTML Forms, ERB (Embedded Ruby) and template rendering, and application environments.

Curriculum Overview

(cont'd)

Topic: Rails

Having a foundation in the Ruby language as well as the architecture of the World Wide Web, you will use Rails to build complex, functional web applications from the ground up. You will learn the file structure of Rails, how to set up your own databases, how to draw routes and create Rails forms, gain an understanding of the asset pipeline, and bring it together by integrating front-end design skills.

You will also have the ability to take on more advanced concepts such as authorization, validation, and callbacks. Once you grasp the basic functionality of Rails, you will spend time building out your own Rails applications, moving through the entire process from idea to execution.

Topic: HTML & CSS

You will master the basic building blocks of how the web is rendered and you will become fluent in the language that makes the web beautiful. Learn how to conceive of and build UIs for your web apps by writing well-structured HTML and CSS, as well as using SASS to create efficient and organized front-ends.

Topic: JavaScript

JavaScript powers the user experience of the web. You will learn the basics of JavaScript syntax, its functional architecture, and different approaches to the object model. You then learn the Document Object Model (DOM) Javascript API provided by the browser to dynamically interact with HTML. This unit focuses on jQuery, the most popular JavaScript library, to aid you in learning how to collect user input, manipulate the DOM with animations and injection, and send Asynchronous AJAX requests for a rich user experience. You will then explore popular JavaScript frameworks including AngularJS, Ember, etc...

Curriculum Overview

(cont'd)

Topic: Node and Express

Built on top of the V8 Javascript Runtime, the Node Javascript ecosystem is becoming a popular and useful tool for asynchronous and real-time application development. This unit focuses on building real-time web application servers with Node and the Express.js framework. You will learn how to build full stack JavaScript web applications from end-to-end with a focus on web sockets and the real-time web.

Projects: Build Real Things

Learn by doing. You will build three to five applications throughout the program. One application will be a robust “Capstone” project, optionally built in a team with other students over several weeks. Projects are an opportunity to explore specific technologies that interest you so you can learn more about technologies for specific industries (ie. students interested in commerce may build applications using shopping carts and payment provider integrations).

Throughout: Being a Software Developer

While the linear progression of the curriculum is focused on technical skill attainment, the track is designed to teach you how to be a software developer. This includes things ranging from honing communication skills (you will be encouraged to maintain an active technical blog and present at technical meetups) to being helpful by answering other Learners’ questions and getting involved in the Learn community across the web.



For more information please check
out our website at Learn.co/verified
or contact us at support@learn.co

