

Online Software Engineering Bootcamp Syllabus & Program Guide



Table of Contents

Program Overview	3
The Power of Learning Ruby and JavaScript	4
Curriculum Overview	5
Full-Stack Web Development	6
Computer Science	7
Portfolio and Projects	8
How We Learn	9
Program Pace and Schedule	10
Contact Us	11

Overview

Flatiron School's Online Software Engineering Bootcamp Program

Flatiron School's proven Full Stack Web Development Curriculum is at the heart of our instructor-led Online Software Engineering Program. This rigorous course of study prepares you for a career in web development by providing you with the skills and experience necessary to obtain and excel in an entry-level software development job. While the bulk of the material covered encompasses the Ruby and JavaScript ecosystems, we carefully designed this curriculum to teach you to think—and build—like a software engineer, independent of any specific language.

Over approximately 800 hours of challenging but rewarding coursework, you will:

- develop a foundation in programming fundamentals
- conquer the concepts of object-oriented programming
- work with APIs (Application Programming Interfaces)
- become proficient in database modeling and ORM (Object Relational Mapping)
- understand MVC (Model-View-Controller), a pattern used by frameworks like Rails to build large-scale applications
- execute application deployment

By the completion of Online Software Engineering Bootcamp, you'll have done much more than build your technical skills: you'll have maintained a technical blog to show you can credibly talk tech; you'll have become a part of your local developer community; you'll have amassed an impressive portfolio of unique, functional web applications that show employers your programming abilities in action.

Proven Results

Flatiron School students get jobs — period. In our last Online Students Jobs Report, 97% of students were hired. As a graduate of our Online Software Engineering Bootcamp, job-seek with the support of our team and you'll get a job offer within 6 months of graduation, or we'll refund your tuition (see eligibility terms). From weekly 1:1 career coaching sessions, to mock interviews, to employer evangelism, our seasoned Career Services team is dedicated to helping our students launch lifelong careers in code.

The Power of Learning Ruby and JavaScript

Why do we teach these languages to aspiring developers?

Readability

Programming is about abstractions and expressions: the mechanics of code are universal and exist in all modern languages. Much of the initial difficulty in learning programming stems from the learning curve necessary to gain comfort with a language's syntax.

Ruby was specifically designed by its inventor Yukihiro Matsumoto to make programmers happy and it's delivered upon that objective: Ruby is expressive, accessible, and reads remarkably like English, allowing new programmers to focus immediately on the fundamental concepts and logic, rather than basic syntax. As such, even beginners can start building right away. We find Ruby to be extremely effective at helping students learn how to think like programmers, break problems down, express themselves technically, abstract ideas, and work together with other programmers.

It's also easy to get started with JavaScript. Programs written in C++ and Java must be compiled to machine code before running. In contrast, JavaScript programs compile at runtime—you *write and it runs*—making it simple to start building.

Career Flexibility

Learning to code and evolving as a programmer is a lifelong endeavor. No matter what language you learn first, you will have to learn other languages throughout your career in order to keep pace with changing technology. Starting off your coding education by learning both Ruby and JavaScript not only makes you a more versatile—and employable—programmer immediately; it also prepares you for the essential task of continuing to learn throughout your career as new languages, frameworks, and technologies are released.

Open Source

Both Ruby and JavaScript have nurtured vibrant, supportive open source communities—there are over 900 Ruby groups on Meetup.com, totaling over 500,000 members worldwide; JavaScript has over 4,000 groups on Meetup.com with over two million members. Ruby also has a huge and useful ecosystem of over 60,000 libraries. This will allow you to leverage free, publicly-available tools to build applications with complexity and real-world use beyond what you could approach otherwise.

JavaScript is Essential for the Modern Web

JavaScript is the language that fundamentally changed the web from static, text-filled pages into the interactive experience it is today—and, as one of the most popular languages out there, it's still bringing websites to life in new, exciting ways. As a front-end developer, you simply can't avoid it. And with Node.js allowing JavaScript to run client- and server-side, it's a valuable tool for full-stack developers as well.

Curriculum Overview

Getting Started

While there are no prerequisites to these programs and we welcome beginners, we do recommend exploring free coding resources (including our free Bootcamp Prep course) before getting started. It's both a way for you to determine if you like programming enough to pursue a career as a software engineer and for us see that you're passionate about learning to code. We want to fill our online community with people who see programming as a craft, people who want to launch lifelong careers as developers.

Mastery-based progression

Labs are taught using test-driven learning, allowing students to gain real-world programming experience by ensuring that all of their tests pass. For Flatiron School's instructor-led programs, this allows our instructors to offer feedback and support students in real-time. Each curriculum section concludes with a comprehensive project meant to demonstrate students' learnings and build a portfolio. Cohort Leads run 1:1 code reviews with students to ensure they've mastered key concepts before progressing to the next section, and are ready to provide additional mentorship if students need help broadening their understanding.

Lifelong Learning

Graduates join an active network of successful software engineers. For Flatiron alumni, engaging with our community doesn't stop at graduation—and nor do opportunities to learn. Flatiron School alumni gain lifetime access to additional curricula on our online platform, Learn.co, including iOS Immersive with Swift and Objective-C, Backend JavaScript with Node.js, and Computer Science.

CURRICULUM OVERVIEW

Full Stack Web Development

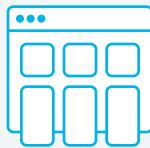
We designed our Online Software Engineering Bootcamp curriculum to give students the necessary expertise in both back-end and front-end programming technologies to become full-stack developers. It's a more extensive course of study than the average school offers – but then we expect more of our students.



Programming Fundamentals

Section 1 HTML& CSS, Ruby, SQL, ORM

After diving into HTML5 & CSS, students get comfortable with object-oriented programming, learning to read websites with Ruby and save data to a database with SQL and Object Relational Mappers.



Web Frameworks

Section 2 & 3 Sinatra, Rails

Students learn two key Ruby frameworks, first mastering the fundamentals of web programming with Sinatra before experiencing how quickly they can build incredible apps with Rails.



JavaScript

Section 4 JavaScript, React

Students gain a thorough understanding of Javascript and functional programming – crucial for front-end devs – and start to build their own version of React before moving onto the framework itself.



Front-end Frameworks

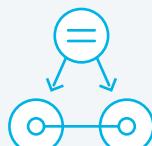
Section 5 & 6 React, Redux, JSON

Students learn to build productive, scalable front-ends with React and Redux, creating slick, functional, reactive code with Redux as a state manager and Rails as the back-end JSON API.

CURRICULUM OVERVIEW

Computer Science

Building complex applications at web-scale requires an understanding of algorithms and Computer Science fundamentals. Studying this discipline gives students an advantage in technical interviews and lays a powerful foundation for students to increase their technical depth throughout their careers.



Algorithms

To understand the resource constraints that affect software performance, students study List Searching and Sorting, Graph Manipulation, and Algorithmic Thinking and Algorithmic Analysis.



Data Structures

To understand the ramifications of how data is stored and accessed, students build their knowledge of Hash Tables, Stacks, Queues, Trees, and Graphs.

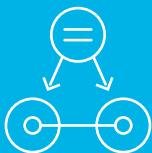


Technical Interview Training

Through extensive practice problems and mock interviews with instructors, students become prepared for the CS component of technical interviews.

Projects & Portfolios

While the linear progression of our curriculum is focused on building technical skills, our aim is to teach students how to become software engineers—which is distinct from simply knowing how to code. Students engage in a number of activities that hone their communication and collaboration skills, helping them build the foundation needed to grow as software engineers in the future.



Portfolio Projects

At the conclusion of each program module, students build advanced Portfolio Projects to demonstrate both the technical skills they've gained in the module and their creativity. Previous projects have won prestigious tech awards, become MVPs for startups, and been presented at the White House. Portfolio Projects represent an opportunity for students to explore specific technologies that interest them while building a portfolio of fully functional web applications to impress employers.



Robust Blog

All Flatiron students maintain technical blogs to show they can not only write code, but communicate how that code works – an essential skill for software engineers.



Active Github Profile

GitHub is the modern software engineer's resume. Students push every line of code they write at Flatiron School to GitHub through our proprietary platform, Learn.co, giving them an extensive profile to show employers and fellow engineers.

How We Learn

Flatiron School's programs are powered by Learn.co, the world's most sophisticated platform for teaching and learning code.

Use Real Tools

You can't learn real skills without real tools. We don't believe in contrived environments or multiple choice quizzes. Learn.co users set up a real development environment with our fast setup process and use a professional command line and Git-based workflow. You'll truly learn by doing.

Open Curriculum

Our industry-tested curriculum has given over 1,200 Flatiron graduates the skills to become web developers and thrive in their careers. And because our curriculum is open-source, it stays more current than any other. Students are encouraged to suggest changes directly from our online platform, and receive public credit for doing so. We continually improve our coursework in reaction to feedback and real-world changes, and our edits are supplemented by hundreds of student submissions each month.

Test Driven Learning

Flatiron School has pioneered a new way to learn that mirrors the ubiquitous practice of Test-Driven Development (TDD), where code requirements are defined before a program is written. Flatiron students complete lessons by writing code that meets requirements established by our curriculum. Tests are automated and descriptive, so students can learn by solving real problems and understanding not only when code is broken, but why.

Program Pace and Schedule

At Flatiron School, we know that how you choose to study is as integral to your success as what you're learning. Paired with our proprietary online learning platform, Learn.co, and individualized support from an Educational Coach, all students have access to a personalized learning experience. Choose from three different program options, each tailored to today's online learner.

	Full-Time (20 person cohort)	Part-Time (40 person cohort)	Self-Paced
Length	5 months	10 months	Up to 15 months
Time Commitment	45-50 hr/week	20-25 hr/week	100% flexible
Admissions	Cultural + Technical Interview	Cultural + Technical Interview	Cultural Interview
Career Services Support	✓	✓	✓
1,000+ Curriculum Hours	✓	✓	✓
Ask A Question	✓	✓	✓
Educational Coaching	2x per month	1x per month	5x package
Live Lectures	✓	✓	✓
Assigned Cohort	✓	✓	
Technical Mentorship	1 hr/week	30 min/week	
WeWork Hot Desk Membership	✓	✓	
Money-Back Guarantee	✓	✓	✓

All courses options include career services support, a money-back guarantee, and provide the same level of rigorous curriculum designed to make you a well-rounded software engineer no matter when you graduate.

Contact us

For more information, please check out our website at www.flatironschool.com or contact us at admissions@flatironschool.com.