Firehose Project

Full-Stack Web Development

Contents

Introduction

Methods

Best Practices

Support

Syllabus

Admissions

Introduction

Welcome

We're excited to have you starting your coding journey with us. Learning to code can be challenging, fun, and extremely rewarding. We're here to help you out along the way and ask in return that you dedicate your time and energy to this program. Doing so will help you get the most out of this program and set you on the path to pursue a new career.

Become a Full-Stack Web Developer

It's a difficult journey to learn how to code. It's going to involve a lot of work. It might be easy for you at times, it might be hard for you at times, just know that everyone has a different experience and certain concepts come easier than others. However, with hard work and focus you can learn to code, just like hundreds of our Firehose members and graduates.

Ruby – The Ruby language was designed with one main principle in mind: developer happiness. It's a programming language that was designed to be easy to read, fast to learn, and simple for developers to use to solve complex problems. The learning curve to master Ruby is less steep in comparison to other programming languages, and after just a short period, you will be able to write and execute Ruby programs and solve complex algorithm challenges. In addition, mastering Ruby first provides you with an ideal springboard to learn another programming language.

The open source Ruby community is massive, very active, and welcoming to people of all skill levels. When your programming craft is backed by an active, worldwide community, it means you have ample opportunities to find and work on interesting projects with awesome people.

JavaScript – Today's user interfaces depend heavily on JavaScript to create a smooth user experience. JavaScript skills are an essential software engineering skill for today and tomorrow. While JavaScript is a powerful and efficient programming language, it's notoriously difficult to learn as a first programming language. Instead, it's perfect for your second language.

In-demand software engineers know multiple programming languages; they're polyglot programmers. When you have experience with more than one programming language, you enable yourself to choose the right tool for the job rather than use the same tool for every job. Having the ability to draw on this flexibility and experience is exactly what will make you in demand.

The following orientation will break down general tips and best practices for learning how to code, as well as the different channels of support that you can use during the Firehose program to get help.

Methods

Exercises

The written curriculum includes most of the information and teaching material. This is how we teach the concepts you'll learn in order to work on the projects, solve coding challenges, and get a job as a developer!

You'll be coding alongside the lessons as you work through them, so you'll be able to use the instruction and compare your results to the expected results outlined in the lessons. If your code or project doesn't look like it should, it's time to put on your problem solving hat! The best developers dive into problems to see where they might have made a mistake or where there's a better solution, but we know you're just starting out so we have a lot of support to help you get unstuck if you need it.

Videos

The video lessons are recordings of short lectures that walk you through high level concepts and overviews of material. These are meant to provide more clarity on complex topics and really help you connect the puzzle pieces. We'll give you pointers throughout the course so you know when it makes sense to watch each video.

Challenges

Throughout the program, you'll work through a handful of challenges to practice what you're learning. Just like a developer on a team in the real world, you'll submit your code for review after completing each challenge. Our team of engineers will review your code and email you with feedback and suggestions for how you can refactor or improve your code.

Quizzes

Quizzes are short activities that allow you to show us what you've been learning and a allow us to evaluate your progress. We'll ask you to complete coding quizzes at specific points throughout our program.

The coding quizzes are meant to make sure you're on track and retaining the material that you should up to this point. If you consistently pass the coding quizzes, know that you're on the right track and well on your way to achieving your goals.

Best Practices

Pay Attention to Detail

Attention to detail is extremely important. When following instructions, it's helpful to always double check that what you're doing is exactly the same as the instructions.

The smallest differences between what the instructions tell you to do, and what you do, can cause major problems. Sometimes it may seem like things are working initially, but not following the instructions carefully could cause problems later on in the course.

So when following a step, review the lesson material and right after you do a step, compare the step you've done with the result and make sure it's exactly the same. It can be helpful to try to look for differences in what you're doing and what the instructions say.

Take Your Time

The Firehose Project has a lot of material. It can be tempting to try to rush through things to complete things. Rushing is bad for a variety of reasons:

First, if you rush through the material too fast you'll likely find yourself making mistakes. Even small typos can cause problems when you're writing code. Fixing these mistakes will often take longer than if you spend time with each step initially and go through the material at a steady pace.

Second, if you skip reading sections, or only skim sections, you will not retain the material as well. We have a variety of Challenge exercises that make sure you've understood important concepts. Rushing through the guided curriculum will often cause you to take more time completing the challenges than you saved from rushing through the material.

In the end, it just doesn't pay off, so it's important to go through the material at a slow and steady pace.

Take Frequent Breaks

Sometimes you might get frustrated. You might see error messages or things might not work the way you expect them to - remember, coding is hard. It's important to put aside anger or frustration to work through the problem logically when you get frustrated. Sometimes taking a break, like going on a quick walk, will allow you to return to the challenge with a clear head and fresh eyes on the problem.

Treat People with Respect

It's important to always treat the people who are helping you out with respect. If you're frustrated about code, it can be easy to forget this fact, but make sure to treat everyone you interact with, with respect. We're real people and our main goal is to help you as best we can with any questions or problems that might arise.

This is also important because in the real world you might be asked tough coding challenges on a technical interview. If/when this happens, you'll want to be in the habit of communicating in a thoughtful, friendly way, even when working with complex coding challenges.

Ask for Help

It's most effective to over communicate about the issues you're facing. The more detailed you can get, the better we can help you solve the issue. Vague questions like, "it doesn't work" are impossible for us to investigate further and provide suggestions. If we don't receive enough information about the issue, it typically requires a lot of back-and-forth which means it'll be longer before you're "unstuck" and back to coding up a storm.

Detailed communication about the issue from the beginning can help reduce the time to solve the issue and prevent confusion or frustration.

In any questions you ask, include all the details:

- What steps have you already done?
- What problem are you seeing?
- Can you include a screenshot?

Make sure to include all these details, and more that might be relevant in your question.

Support

1-on-1 Training

Whether it be questions on your career path, or technical concepts, your trainer is here to coach you through it all.

Personal 1-on-1 trainer sessions are a fundamental part of the Firehose experience and complement our core coding curriculum. Together with your trainer you will accelerate your web development skills and learn how to work with an experienced programmer.

Your trainer will work regularly with you 1-on-1 to help you reach your coding goals. You trainer will structure your sessions around your personal needs and goals, and you can expect them to challenge you, assign additional work to you, and guide you through the complex world of professional web development.

Trainer Pairing – We personally match you with the trainer who is the best fit for you based on several variables, including your personal coding goals, your technical background, and your performance in the prep course.

We'll email you an introduction to your trainer during your first week in the program so you can schedule your first trainer session.

Session Format – Each trainer session is one hour long. Generally speaking, your trainer will structure each of your sessions for maximum impact. This means you'll usually spend the first 10 minutes going over your list of questions, then take the remainder of the session to work on additional topics that are designed to help you reach your coding goals. Often times, that includes pair programming, going through your algorithm code, and building out additional features for your web application.

Session Preparation – Trainer sessions can push your knowledge to the next level and allow you to learn additional skills. This can only happen if you arrive prepared. This means you should have worked on your web application in the days leading up to your trainership session and have your assignments completed. You should also make sure you have a list of high-level questions ready and reviewed.

Session Schedule – You should schedule sessions at a time that is convenient for both you and your trainer, and it is very important that you schedule sessions consistently and avoid letting multiple weeks go by without a trainer session.

Session Arrival – You need to arrive on time to every trainership session. If you are not in the video hangout within the first 5 minutes of your scheduled session, your trainer will reach out via email to check in with you. If they don't hear anything back, they will assume you've opted to miss this session and will officially disconnect the video chat 15 minutes after the scheduled starting time.

Session Extensions – Trainer sessions are intentionally scheduled to last for 1 hour. Sometimes, trainers volunteer to go over the standard hour time slot to ensure you are all set on the right track and ready to go. However, this is at the discretion of each individual trainer and it's not uncommon for your trainer to have another session scheduled immediately after your session ends.

Be respectful of your trainer's time and be cognizant of how you're spending your session time. To ensure all your questions get answered and that you leave the session all set to continue learning, ask your biggest and most important questions at the beginning of your trainer session and not 5 minutes before the end of your session.

Session Rescheduling – Your trainer prepares for each of your sessions in advance and has dedicated their time to helping and teaching you. If you can't make it to your scheduled trainership session, please let your trainer know at least 6 hours in advance of your scheduled session. If you miss the 6-hour mark, your trainer will count your session in full. (This will count as your one "excuse wildcard" if you haven't used it.)

Session Absences – We know life can be unpredictable, and you shouldn't have to miss out on a full trainer session because of something beyond your control. That's why we grant one "excuse wildcard" per member that you can use if you have to miss a trainership session due to an emergency.

If you miss scheduling a session at any point, you should work with your trainer to make up that session as soon as possible.

If your trainer isn't in your video hangout within the first 5 minutes of a scheduled session, please reach out to them via email to check in. If you don't hear back from them within the first 15 minutes of your scheduled session, or cancels a session less that 6 hours before the scheduled start time, email us at hilary@thefirehoseproject.com to let us know.

Unused Sessions – All trainer sessions must be used before your graduation date. Unused sessions will expire when you graduate.

Technical Feedback

This is where we teach you to think like a developer. We'll walk through your code, and your process to help you think about problem solving in the right way.

When you complete a project or challenge, you'll submit it through the Firehose platform for personal code review from the Firehose team. Code reviews help you learn with constructive feedback and improve your coding as you continue through the course. As a developer, you'll receive frequent code reviews from your team, so it's a great opportunity to practice receiving and implementing feedback.

Immediate Support

When you're stuck on a problem, every course comes with immediate support to keep you moving in the right direction.

The technical forum at the bottom of each lesson is available to provide specific steps to fix specific problems in the project that you're working on. The forum is designed to quickly get you "unstuck" and back on track so you can continue coding up a storm.

Asking Questions – Before posting questions in the forum, you should try to work for 20-30 minutes to solve problems on your own. This helps you prepare to get to a self-sufficient developer status, which is so important because after graduation from the accelerator, you want to make sure you have the skills to code and solve problems on your own.

When you do post a question in the forum, make sure to read each reply carefully and double check you're doing things that are asked or suggested. Remember, be detail oriented! Skipping steps in the forum answers will cause more back-and-forth and ultimately take longer to get you "unstuck."

Getting Answers – We generally check the forum several times each day and answer all questions. To get a fast answer, be thoughtful and detailed in writing your question and explaining your problem.

Since we have a small team supporting all of the members in the forum, you can expect specific step-by-step instructions to get unstuck with the problem at hand, but most learning should be done via other channels.

Keep in mind that Firehose trainer sessions, office hours, and the Firehose Slack channels are great ways to discuss more complex topics and continue learning outside of the written and video lessons.

Complex Questions – We recommend saving your bigger concept and high-level questions for your trainer and using the forum if you need help troubleshooting a specific, time-sensitive error. There is always a dedicated team member providing support in the forum, so for time-sensitive help, the forum is much faster.

Member Support

You don't have to go through it alone. Find fellow career shifters who are on the same journey and connect with people who get it.

Once you begin the full accelerator, you'll be invited to join our Firehose Slack channels. Slack is a tool that developers use to communicate whether they're working on projects in the office together or coding on a remote team.

Our member Slack is very active and a great place to chat with other members, seek help, and help others out. We recommend being logged into your Firehose Slack account whenever you're coding. Participating on Slack is a great way to meet some of your fellow Firehose members, graduates, and trainers.

Our member Slack has a #Coding-Help channel, where members ask questions when they're stuck and help each other out. Asking questions in this channel is a great way to get help and insight from your Firehose peers.

Syllabus

Full-Stack Web Development

The best way to retain knowledge is by doing. Our Program will teach you to think like a developer and problem solve like one. We've taken years of on-the-job learnings and transformed them into deliberately constructed lessons to teach members the skills they need to enter an actual web developer role.

Phase I – Build

In this phase, we will work together to build your foundational skill set.

1. Intro to Software Engineering
2. Technical Immersion
3. Development Process and Structure

Phase II – Apply

Now that you have your foundation, we will apply your skills to make you employable.

4. Employable Experience
5. Technical Evaluation
6. Developer Validation

Phase III – Execute

We will take all of your skills and apply them within a real world setting.

7. Real Team Experience
8. Showcase Yourself

Intro to Software Engineering

Duration: 15 – 30 hours
Contents: 43 steps, 5 challenges

Learn the Fundamentals
You'll work with us to build multiple scripts in Ruby. We'll provide you the technical feedback as you learn how to write programs, manage flow control, and use Ruby methods. At the end of this course, you'll have a live portfolio website to showcase your work.

Technical Immersion

Duration: 15 – 30 hours
Contents: 19 steps, 11 videos, 2 assignments

Use the Tools
You'll dive in the deep end by building a database-driven application using the Rails framework. You'll learn how to manage version control using real developer tools, and host an application online. At the end of this course, you'll have a custom quote generator to showcase your newly acquired skills.

Development Process and Structure

Duration: 90 – 105 hours
Contents: 51 steps, 4 challenges, 12 videos, 1 quiz

Refine the Skills
You'll learn how to use algorithms, object-oriented programming, and APIs, which employers will look for when hiring web developers. You'll data structures this knowledge to build features like user authentication, comments and ratings, and image uploading.

Employable Experience

Duration: 120 – 135 hours
Contents: 39 steps, 2 challenges, 34 videos, 2 quizzes

Build a Video Streaming Platform
You'll take your foundational skills and take them to the next level. You'll learn how to design complex user interfaces, architect advanced database relationships, and link it all together with by solving challenging algorithms. At the end of this course you will have developed a custom marketplace with payments.

Technical Evaluation

Duration: 90 – 105 hours
Contents: 21 steps, 5 challenges

Test the Skills
You'll learn how crucial testing is to the development process. You'll build automated software tests, ensure good test coverage, and release tested code at an advanced level. At the end of this course, you'll will have grasped the test-driven development process.

Developer Validation

Duration: 105 – 120 hours
Contents: 20 steps, 9 challenges

Apply the Knowledge
You'll take the skills you've learned and apply them to a completely different programming language to showcase your versatility as a developer. At the end of this course, you will build an entire dynamic application through JavaScript.

Real Team Experience

Duration: 120 – 150 hours
Contents: 3 steps, 8 sessions

Prove You're a Developer
You'll transition from individual developer to contributing team member. You'll work with other members and a scrum master, meeting weekly to discuss objectives and divide tasks. During this course, you will build a complex chess app using the agile methodology to demonstrate to employers that you can operate within a professional work environment because you've worked with other developers in a meaningful way on a complex project.

Showcase Yourself

Duration: 135 – 150 hours
Contents: 52 steps, 17 challenges, 2 downloads

Conquer the Job Search
You'll now be fully-equiped to change careers and we're here to help you get there. We'll guide you through the current job landscape and work together to craft a true-developer resume, write a cohesive cover letter, and present your work in the best light possible. This will help you develop your unique story to stand out to employers.

Admissions

Program Breaks

Keep in mind that taking a break will affect your learning. We generally encourage members who decide to take a break to only take as much time as is necessary, and we only allow each member to take one break. This is because of the additional time and effort it takes to get back to your previous level of coding after a break.

If you need to take a break from the program for any reason, we have the below two options available.

Option 1: Pause – A pause is a short break of less than two weeks. Your graduation date will be adjusted according to the length of your pause and once you return, you will work with your same trainer. During your pause, you will not have access to the lessons, resource videos, or coding challenges.

Option 2: Freeze – A freeze is a break that is between two weeks and six months long. A freeze can begin on the date of your choosing and will end on either a date of your choosing or six months after your freeze begins, whichever occurs first.

Once you have chosen your return date, your graduation date will be adjusted according to the length of your freeze. We will try to match you with your same trainer after your freeze, but we cannot guarantee their availability. If necessary, we will assign you to a new trainer. During your freeze, you will not have access to the lessons, resource videos, or coding challenges.

To end your freeze, contact your Trainer at least one week in advance of your chosen return date.

Program Extensions

All program extensions include continued access to office hours and use of the forum. Please contact your Trainer to determine which option best fits your needs.

Program Plans

If you have any questions or concerns about your current pacing plan or are interested in switching plans, please contact your Trainer.

Leaving the Program

We try our best to meet the needs of all of our members, but we understand that sometimes members need to leave the program. If you need to leave the program for any reason, please email brita@thefirehoseproject.com.

Our refund policy depends on the length of time a member has spent in the program. For the purposes of determining your refund, the date we consider you to leave the program is the date that you email Brita your decision to leave.