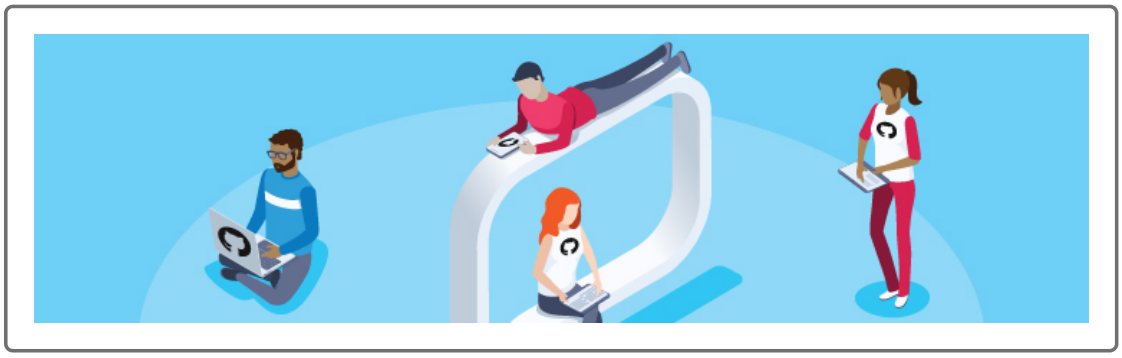
7 Introduction to Module 7



Congratulations! You've reached a significant milestone in this course, as you now have a firm grasp on all of the foundational technologies you need to become a full-stack web developer.

Before you get started on this module's work, you may have noticed that this module looks different from the previous ones you've worked on. Don't be alarmed by this! Over the next two weeks, you'll be working on a group project, and these modules are structured a little differently. It's important to read through all of the pages in this module to understand the expectations for the next two weeks.

Think Like a Developer

This course is divided into three equal phases, including six weeks of content and two project weeks. You've just completed the content for the Foundation phase, giving you a solid foundation in the fundamental concepts of web development. You've also learned what it takes to think like a developer, and you've been practicing the habits that you'll need to succeed in your career.

Git Workflow and Version Control

Working through the previous six modules of content has shown you how you should approach each new project you work on. Using Git is the industry standard for developers, and you'll use it in any role you take on. You've also used Git for version control, which is extremely important in the development world. The ability to track changes over time and roll back to previous versions when necessary is absolutely vital to coding at scale.

Collaborative Coding

GitHub is the largest collaborative coding platform in the world. You're now comfortable with creating repositories for your projects, creating feature branches, and working with existing code. When you learned about server-side APIs, you worked closely with a fellow developer, Amiko, to complete a project. You will find yourself in this very situation over and over as your career as a developer continues.

Creating Milestones with GitHub Issues

Speaking of GitHub, you've also learned that developers work with a narrowly defined set of tasks, known as issues, that can include bugs that need to be addressed, feature requests, or other milestones that you'll need to complete to make progress with your project or application. This is how you will report errors in the codebase you use as a developer, and it is how you'll receive assignments for tasks that you need to complete.

Agile Project Management

One of the most important skills you've started learning over the last six weeks is agile project management, a popular management style used at most development companies. Although you'll learn more throughout the remainder of this course, you have already approached each of your Challenge assignments through the lens of agile development, working with user stories, acceptance criteria, mock-ups, and code review. You'll encounter these in almost every job in web development.

Pseudocoding

For developers, opening a code editor is similar to a writer staring at a blank page. You know where you need to start and you know where you want to end, but how do you get there? Pseudocoding helps you get from that starting point to the end goal by simplifying the structure of your code into human-readable steps. After you've defined these steps in simple language, you can translate them into the individual sections of code you'll need to create your program. Every developer in the world, from beginners to senior programmers, uses this process to solve problems and create code, and that includes you!

Reading Technical Documentation

Unlike some professions, web development isn't just about the knowledge that you have on any given subject. It's also about your ability to navigate new technologies and dive into the documentation provided to you. Every time you've used a walkthrough in this course, in the lessons or in each module's Up and Running section, you've practiced this vital skill. Employers are looking for developers with a certain set of skills, but they're also looking for people who are able to solve problems and adapt to learning new skills.

Interview Preparation

As we mentioned previously, employers are looking for developers with problem-solving skills. One of the most common ways they test these skills is through technical whiteboarding interviews, where potential candidates are asked to solve a particular algorithmic problem on a whiteboard. Getting these problems "right" is only part of what they're

looking for. They also want to see how you get to your answer. The algorithms you've practiced as part of your Career Connection are setting you up for success once you start interviewing for roles.

Career Preparation

Networking and behavioral interviewing are the other important skills you'll need to practice to get you prepared for your first development role. The Career Connection is designed to help you practice your soft skills—the personal, social, and communication skills you'll need for success on the job. You're also gaining some context on how the technical skills you're learning fit into the various roles available for developers so that you can start to think about which area you'd like to specialize in.

Lifelong Learning

The most telling trait for all developers is their desire to keep learning long after their education has ended. That is why you're in this course to begin with, so you're on the right track. Every time you see a Deep Dive, Nerd Note, or Legacy Lore callout; look into some of the resources we've provided in the Dessert Menu; or reflect on the week you've had and write down your answers, you're practicing what all developers do for the rest of their lives: learning! This course teaches you how to hone those skills. It's up to you to make it part of your routine.

Everything You've Learned

You've covered a lot in the past six weeks. Let's take a moment to recap all that you've learned.

In Module 1, you learned about the fundamental building blocks of the web—HTML and CSS—navigated your computer with the command line, learned about tracking changes to your code using Git and GitHub, and deployed your code to GitHub pages.

In Module 2, you made your applications mobile responsive and learned how to use advanced CSS, including flexbox, CSS Grid, and CSS variables, to make your websites look better.

In Module 3, you took your first steps as a programmer by learning the third building block of the web, JavaScript. You created your first JavaScript functions, learned about data types, and used conditional statements and iteration.

In Module 4, you worked with your first APIs, using the web APIs native to the browser to manipulate and traverse the DOM, handle events such as key presses and clicks, and use the `window` object to store data on the client side.

In Module 5, you integrated third-party APIs into your own applications, using the JavaScript library jQuery to simplify DOM manipulation and the CSS framework Bootstrap to quickly implement a component-based grid layout.

Finally, in Module 6, you worked with the back end for the first time, using the Fetch API to make requests to server-side APIs and integrating the returned data from third-party servers into your own applications.

You Are a Developer!

You are now a developer! Over the rest of this course, you'll learn about back-end development and ways to improve the performance and efficiency of your applications, but in six short weeks, you have already learned all of the fundamental technologies used in every website in the universe. Every single website you visit uses HTML, CSS, and JavaScript as its building blocks. Every website and application that you use that deals with data uses APIs to receive and send that information.

After a month and a half, you are already building applications that look very similar to the websites you encounter in your daily life. You're creating user interfaces with hover effects, click events, and forms that take input

from users. You're hooking into APIs to display data to the user. You're creating layouts that respond to multiple screens and sizes. Take a moment to visit some of the sites that you use every day and notice the things that you are now able to build.

While there is certainly more to learn, you now have the skills to work on the front end of any website.