3.1.1 Introduction

If you've ever accomplished anything useful in a web browser, you can thank JavaScript. While HTML and CSS define the structure, content, and layout of a webpage, JavaScript lets visitors "do things," such as fill out and submit a contact form, click a button to open a dialog window, and add or remove content without refreshing the browser.

JavaScript is a powerful, wildly popular, general-purpose programming language. Some notable examples of web apps written in JavaScript include Netflix, GitHub, YouTube, and Facebook.

One reason for JavaScript's popularity is that it's relatively easy to learn. Another is that all the major web browsers can run JavaScript code. Front-end web developers who know JavaScript—like you in the near future!—can create powerful, dynamic web applications with rich user interfaces.

NERD NOTE

JavaScript is used on 95% of the almost 2 billion websites in the world today!

It's worth noting that most useful web applications communicate with back-end applications running on internet-connected servers that can do heavy data processing and store large amounts of data. JavaScript works there, too! We'll learn more about creating back-end (often called **server-side**) applications in future modules. For now, rest assured that what you learn in this module will set the foundation for the remainder of the course.

Let the Games Begin

In this module, we'll use JavaScript to build a browser-based video game to enter in the Con Solo Game Jam, a competitive hackathon.

Game jam participants must create an engaging game that meets the following requirements:

- The game must be created by a single developer in 24 hours.
- The game must run in a web browser.
- The game must use the provided `index.html` file.
- Only the `<title>` element of `index.html` may be changed.
- Participants cannot use CSS.
- All game code must be contained in the `game.js` JavaScript file.
- Extra points for code quality!

After some reflection, we've decided to build an action-packed robot combat simulator game called Robot Gladiators. During the game, the player will coach their robot through a series of fights, gaining cash, attack power, and repairs along the way. The robot who survives with the most cash will be remembered in the browser's storage system! While we begin the build of the game we will be covering the following JavaScript fundamentals as a part of this lesson's objectives:

- Use of functions to communicate with the user.

- Use variables and operators to assign and manipulate data.
- Manage conditional statements to control the flow of the application.

First Things First

Because we're on a tight deadline, we need to focus. We'll prioritize the development tasks necessary to ship an **MVP**, or **minimal viable product**. An MVP is the simplest version of a product that a developer can share with target users to get important feedback. Once we have our MVP, we'll share it with other game jam participants and get their feedback.

Our old friends HTML and CSS are on the bench in this round, but a lot of their concepts will come into play while we learn JavaScript, such as the following:

- We wrote HTML in an `index.html` file, and CSS in `style.css` file. We'll write JavaScript in a `game.js` file.
- HTML and CSS have structure and syntax rules, as does JavaScript.
- We used Chrome to load and render HTML and CSS. We'll use Chrome to load and run JavaScript, too.
- We used Chrome DevTools to work with HTML and CSS. We'll also use Chrome DevTools to work with JavaScript.

And as usual, we'll use GitHub issues and branches to manage our work.