Begin to Code with JavaScript

Rob Miles

This is a pre-release section of the work “Begin to Code with JavaScript”. Everything is subject to change, especially the jokes. You can find the “Begin to Code with JavaScript” podcast page here:

www.robmiles.com/jspodcast

The code samples for the book and links to the screencasts can be found here:

www.begintocodewithjavascript.com

Feel free to send constructive comments to writing@begintocodewithja-vascript.com

Introduction

Programming is the most creative thing you can learn how to do. Why? If you learn to paint, you can create pictures. If you learn to play the violin, you can make music. But if you learn to program, you can create entirely new experiences (and you can make pictures and music too, if you wish). Once you've started on the programming path, there's no limit to where you can go. There are always new devices, technologies, and marketplaces where you can use your programming skills.

Think of this book as your first step on a journey to programming enlightenment. The best journeys are undertaken with a destination in mind, and the destination of this journey is "usefulness." By the end of this book, you will have the skills and knowledge to write useful programs and make them available to anyone in the world.

But first, a word of warning. I would not say that learning to write programs is easy. This is for two reasons:

- If I tell you it's easy, and you still can't do it you might feel bad about this (and rather cross with me)
- If I tell it's is easy and you manage to do it, you might think that it isn't worth doing.

Learning to program is not easy. It's a kind of difficult that you might not have seen before. Programming is all about detail and sequencing. You must learn how the computer does things and how to express what you want it to do.

Imagine that you were lucky enough to be able to afford your own personal chef. At the start you would have to explain things like "If it is sunny outside I like orange juice and a grapefruit for breakfast, but if it is raining I'd like a bowl of porridge and a big mug of coffee". Occasionally your chef would make mistakes, perhaps you would get a black coffee rather than the latte that you wanted. However, over time you would add more detail to your instructions until your chef knew exactly what to do.

A computer is like a chef that doesn't even know how to cook. Rather than saying "make me a coffee" you would have to say, "Take the brown powder from the coffee bag and add it to hot water". Then you would have to explain how to make hot water, and how you must be careful with the kettle and so on. This is hard work.

It turns out that the key to success as a programmer is much the same as for many other endeavors. To become a world-renowned violin player, you will have to practice a lot. The same is true for programming. You must spend a lot of time working on your programs to acquire code-writing skills. However, the good news is that just as a violin player really enjoys making the

instrument sing, making a computer do exactly what you want turns out to be a very rewarding experience. It gets even more enjoyable when you see other people using programs that you've written and finding them useful and fun to use.

How this book fits together

I've organized this book in three parts. Each part builds on the previous one with the aim of turning you into a successful programmer. We start off discovering the environment in which JavaScript programs run. Then we learn the fundamentals of programming and we finish by making some properly useful (and fun) programs.

Part 1: The JavaScript world

The first part gets you started. You'll discover the environment in which JavaScript programs run and learn how to create web pages containing JavaScript programs.

Part 2: Coding with JavaScript

Part 2 describes the features of the JavaScript that you use to create programs that work on data. You will pick up some fundamental programming skills that apply to a wide range of other languages, and get you thinking about what it is that programs actually do. You'll find out you how to break large programs into smaller elements and how you can create custom data types that reflect the specific problem being solved.

Part 3: Useful JavaScript

Now that you can make JavaScript programs it's time to have some fun with them. You'll discover how to create good looking applications, how to make programs that are secure and reliable and finish off with a bit of game development.

How you will learn

In each chapter, I will tell you a bit more about programming. I'll show you how to do something, and then I'll invite you to make something of your own by using what you've learned. You'll never be more than a page or so away from doing something or making something unique and personal. After that, it's up to you to make something amazing!

You can read the book straight through if you like, but you'll learn much more if you slow down and work with the practical parts along the way. Like learning to ride a bicycle, you'll learn by

doing. You must put in the time and practice to learn how to do it. But this book will give you the knowledge and confidence to try your hand at programming, and it will also be around to help you if your programming doesn't turn out as you expected. Here are some elements in the book that will help you learn by doing:

Make Something Happen

Yes, the best way to learn things is by doing, so you'll find "Make Something Happen" elements throughout the text. These elements offer ways for you to practice your programming skills. Each starts with an example and then introduces some steps you can try on your own. Everything you create will run on Windows, macOS, or Linux.

Code Analysis

A great way to learn how to program is by looking at code written by others and working out what it does (and sometimes why it doesn't do what it should). The book contains over 150 sample programs for you go look at. In this book's "Code Analysis" challenges, you'll use your deductive skills to figure out the behavior of a program, fix bugs, and suggest improvements.

What Could Go Wrong?

If you don't already know that programs can fail, you'll learn this hard lesson soon after you begin writing your first program. To help you deal with this in advance, I've included "What Could Go Wrong?" elements, which anticipate problems you might have and provide solutions to those problems. For example, when I introduce something new, I'll sometimes spend some time considering how it can fail and what you need to worry about when you use the new feature.

Programmer's Points

I've spent a lot of time teaching programming. But I've also written many programs and sold a few to paying customers. I've learned some things the hard way that I really wish I'd known at the start. The aim of "Programmer's Points" is to give you this information up front so that you can start taking a professional view of software development as you learn how to do it.

"Programmer's Points" cover a wide range of issues, from programming to people to philosophy. I strongly advise you to read and absorb these points carefully—they can save you a lot of time in the future!

What you will need

You'll need a computer and some software to work with the programs in this book. I'm afraid I

can't provide you with a computer, but in the first chapter you'll find out how you can get started with nothing more than a computer and a web browser. Later you'll discover how to use the Visual Studio Code editor to create JavaScript programs.

Using a PC or laptop

You can use Windows, macOS, or Linux to create and run the programs in the text. Your PC doesn't have to be particularly powerful, but these are the minimum specifications I'd recommend:

- A 1 GHz or faster processor, preferably an Intel i5 or better.
- At least 4 gigabytes (GB) of memory (RAM), but preferably 8 GB or more.
- 256 GB hard drive space. (The JavaScript frameworks and Visual Studio Code installations take about 1 GB of hard drive space.)

There are no specific requirements for the graphics display, although a higher-resolution screen will enable you to see more when writing your programs.

Using a mobile device

You can run JavaScript programs on a mobile phone or tablet by visiting the web pages in which the programs are held. There are also some applications that can be used to create and run JavaScript programs but my experience has been that a laptop or desktop computer is a better place to work.

Using a Raspberry Pi

If you want to get started in the most inexpensive way possible you can use a Raspberry Pi running the Raspbian Operating System. This has a Chromium compatible browser and is also capable of running Visual Studio Code.

Sample Code

In every chapter in this book, I'll demonstrate and explain programs that teach you how to begin to program—and that you can then use to create programs of your own. You can download this book's sample code from GitHub by following the link here:

http://www.begintocodewithjavascript.com/code

GitHub was developed as a software development platform but it has turned out to be much

more than that. It is a place where anyone can store files and share them. All the sample programs used in the book are held on GitHub.

At the start of the book you'll discover how to use GitHub to make your own copy of the sample programs. You can then use GitHub to publish JavaScript enabled web pages for anyone in the world to view.

You will need to connect to the internet and create a GitHub account (it is free) to do this.

Electronic media

For the important content elements, I've made some videos. The book text will contain screenshots that you can use, but these can go out of date. Follow the links to the walkthroughs to get the latest steps to follow. There are also some audio recordings you can listen to if you feel brave. You can find all these here:

http://www.begintocodewithjavascript.com/media

Acknowledgments

Thanks to everyone for giving me another chance to do this kind of thing.