Learn JavaScript with p5.js

Coding for Visual Learners

Engin Arslan

Apress®

Learn JavaScript with p5.js

Coding for Visual Learners

Engin Arslan

Apress®

Learn JavaScript with p5.js: Coding for Visual Learners

Engin Arslan
Toronto, Ontario, Canada

ISBN-13 (pbk): 978-1-4842-3425-9 ISBN-13 (electronic): 978-1-4842-3426-6
https://doi.org/10.1007/978-1-4842-3426-6

Library of Congress Control Number: 2018935139

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Natalie Pao
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-3425-9. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Author

Engin Arslan is a Software Developer with a Bachelor of Science in Materials Engineering and a Postgraduate Degree in Visual Effects. Before becoming a Developer, he worked as a Visual Effects Artist / Technical Director on films and TV shows including *Resident Evil*, *Tron*, *Mama*, *Pompeii*, *Vikings,* and *Strai*n. He received an Emmy nomination and won a Canadian Screen Award for his achievements in Visual Effects. During his time in VFX, he fell in love with Python and with programming in general. As a result, he changed careers to be able to immerse himself completely in software development. Engin currently works at a Toronto-based digital services company, where he helps develop solutions in strategic problem spaces using emerging technologies. He also works at Seneca College as a part-time professor and creates online courses for Lynda/LinkedIn and Pluralsight.

About This Book

The emphasis of this book will be primarily on learning programming using JavaScript and p5.js and secondarily in creating visuals. The main focus is to teach you how to program so that you can choose to pursue whatever field that you would like with your newly established skill set. The skills that you will acquire from this book are highly transferrable and can be used with whatever you choose to build: whether web applications, programmable robots, or generative art. This means that I will provide you with enough context so that you can build a strong foundation for programming. But I also won't hinder your momentum with irrelevant technical or theoretical points. The aim is to build a strong but a minimum viable knowledge to get you running with coding. This is the book that I wished I had available when I was learning coding myself.

If you are an artist or a visual designer, this book is perfect for you as you might find the examples we will be building to be directly relevant to your work. If not, this is still a great book for learning programming as the visual nature of the exercises will help you grasp the fundamentals of programming more easily and let you build a strong foundation in a shorter amount of time.

This book will present various JavaScript and p5.js features and concepts in the following chapters. The knowledge will be reinforced by building several useful examples like an animation and a data visualization; and as a final project, we will be building a game that can be deployed online using what we learned in this book!

Here is a rundown of the topics that we will be covering:

Chapter 1 - Introduction: Provides an overview of coding versus programming.

Chapter 2 - Getting Started: We will learn some very basic JavaScript commands and operations to get started with using p5.js.

Chapter 3 - Colors in p5.js: This will be a p5.js-specific chapter where we learn about how colors are defined and used in p5.js. This doesn't pertain to JavaScript but needs to be explored regardless to be able to use p5.js in a comfortable manner.

Chapter 4 - Operations and Variables: We will make use of the JavaScript knowledge we acquired in the second chapter in p5.js context.

Chapter 5 - Conditional Statements and Comparison Operators: This chapter will allow us to write programs that can respond to different conditions by using conditionals and comparison operators.

Chapter 6 - More p5.js Variables: This will be another p5.js-specific chapter where we will learn about several library-specific variables.

Chapter 7 - Loops: Here we will learn about loops, which will allow us to build programs that handle enormous amounts of calculations.

Chapter 8 - Functions: Functions are the building blocks of JavaScript and we will learn more about them in order to build more scalable, modular, and robust programs.

Chapter 9 - Objects and **Chapter 10 - Arrays**: Objects and Arrays are JavaScript data structures that will allow us to organize our code and handle complexity in more intelligent ways.

Chapter 11 - Events: Event handling will allow us to write programs that handles user interaction.

Chapter 12 - More on p5.js: Another p5.js-only chapter where we learn more about library-specific features before diving into our final project.

Chapter 13 - Final Project: We will build a game using everything we have learned up to this chapter!

CHAPTER 1

Introduction

At this age and time that we live in, coding is simply invaluable. It has the power to uplift your career, your future prospects, and even your intellectual capacity. Computation is driving one of the largest capital expansions in history, and there has never been a better time to learn coding than now.

Why Learn Coding?

My first serious interaction with coding was at college. We had to take a course on a programming language called C Sharp. I failed the course the first time I had to take it and barely passed it the second time when I had to take it again. With that defeat in mind, I stayed away from coding for the longest time. I considered it to be a talent that I simply didn't possess. Later, I went on to change my career from engineering to visual effects as I wanted to work in a field that had more room for creative expression. But working in visual effects, I came to realize that the entire operation is actually enabled by the power of computation. From the software that is used to the pipeline management that facilitates the production… Coding is everywhere. It allows studios to deliver mind-blowing effects for movies that make hundreds of millions of dollars in the box office.

Upon realizing the power of coding in my field, I decided to embark on a journey to learn more about it. I ended up teaching myself Python, a programming language that is widely used in visual effects. And doing

E. Arslan, *Learn JavaScript with p5.js*, https://doi.org/10.1007/978-1-4842-3426-6_1

so has been immensely gratifying. Not only has it allowed me to become more accomplished in my work in visual effects and create award-winning effects, but it has also empowered me to transition to an even more rewarding career in software development.

Coding vs. Programming

You must be hearing the terms coding and programming in similar contexts and might be wondering what the difference between them is. In the past few years, coding has become the term of choice to make programming more approachable to the general population. Basically the premise is that you could be coding and still be contributing to the digital economy without actually doing programming.

Let me give you an example of that: you could be using web languages such as HTML and CSS, which are not programming languages. So when coding in those languages you are not really programming but styling or structuring websites (more on their usage in the next section). But you could also be coding in JavaScript, which is an actual programming language. Programming languages allow you to make a computer "do" things. Every time you are programming something, you are also coding. But when you are coding, you might not be programming. Coding is a more general term that is used for describing all cases where you are communicating intent to the computer.

Basically you can think of programming as a subset of coding. But if truth be told, these two terms are used almost interchangeably nowadays. The main purpose of this book is to teach you how to program. We will be coding for programming purposes by using the programming language JavaScript.

On HTML and CSS

Looking at my path for learning programming, I find some of the efforts to teach coding to beginners to be a bit lacking. One of the primary problems in the area is using HTML and CSS as introductory languages.

The problem with these languages is that they are not even programming languages! HTML is a markup language that is used to define the structure of a document in a way that a web browser would understand. For example, HTML teaches you how to write text for a browser so that the browser would know what parts of it is a document header vs. a paragraph, etc…

Likewise, CSS is not a programming language either. It is a styling language that allows us to style HTML documents to have them look aesthetically pleasing and ideally make them more user friendly than before. Furthermore, even though CSS can be used to create incredibly good looking results, it is usually very unintuitive to work with and can be hard to reason about even for a programmer. Learning CSS, you are not only not learning programming, you are very likely engaging in an activity that might not be fun as a beginner if styling websites is not your sole intention.

This push to teach coding using these languages is understandable. After all, given the large dominance of web applications and their immense profitability in certain cases, people found themselves wanting to build their own projects for the Web. And if you are to build a website, you need to use these languages to a certain degree. But having these languages as a starting point could create a misconception about what coding is. Coding can be an immensely rewarding and engaging activity when you are building programs or applications as the domain of possibilities is substantially bigger. As discussed previously, we need to be using programming languages to build programs so the apparent question is: “What makes a language a programming language?”

You can always check Wikipedia for a semi-formal definition. But to me, for a language to be considered a programming language, it needs to have certain control structures available to it that would allow us to express some basic operations. Even this definition probably makes little sense to a beginner. What is meant is that there are structures in programming languages that allow the computer to perform *logical* operations. Some of the examples of such structures, which we will see more about later, are the following: conditionals that allow the program to output different results based on given conditions and variables that store values or loops that allow a program to repeat operations for a desired amount of time. Don't worry if none of this makes any sense right now; the purpose of this book is for us to learn about all these fundamental programming concepts.

Almost all programming languages have these kinds of basic structures that enable us to construct immensely more complicated applications. Think of English, or any other language you might know. You have verbs, nouns, and adjectives. And using these building blocks, people can say the simplest things or go on to write amazing novels. And these are the building blocks that are missing from HTML and CSS that make people miss out on what could be achieved when using programming languages.

In this book we will learn all these basic structures that would allow us to communicate our intent to the computer using the programming language JavaScript.

Why Learn JavaScript?

There are many programming languages out there. This book will be teaching you how to code, by using the immensely popular programming language JavaScript.

JavaScript is one of the most widely used programming languages out there as it is built into every web browser. Due to this, almost all the web pages and applications out there use JavaScript to some degree. In recent

years JavaScript started to be used not only to program user interaction in web pages but also server side - back-end - applications, Internet of Things (IOT) devices or mobile apps for platforms such as Android or iPhone. Even though it has its roots in web development, JavaScript knowledge is now applicable to a vast number of other domains.

Given the popularity and ubiquity of JavaScript, it is really easy to find resources and information about it if you are to ever get stuck. It has a big, vibrant community behind it. In the popular Q&A website, StackOverflow, there are more than a million questions that are related to JavaScript. If you end up coding in this language and get stuck in a problem, the chances are that someone else also had the same problem, posted a question on this website, and got an answer that you can learn from.

I won't go into details of what makes a programming language dynamic or static, but being a dynamic programming language, JavaScript code is more concise and easier to write compared to static languages. Listings 1-1 and 1-2 are some examples where a simple statement that displays the words 'hello world' to the screen are written by using different languages. Notice how much shorter it is to write the same code using JavaScript.

Listing 1-1. Displaying Hello World to the screen in C++ (Source: `http://helloworldcollection.de/`)

```
// Hello World in C++ (pre-ISO)
#include <iostream.h>

main()
{
        cout << "Hello World!" << endl;
        return 0;
}
```

Listing 1-2. Displaying Hello World to the screen in Java (Source: http://helloworldcollection.de/)

```
// Hello World in Java

class HelloWorld {
        static public void main( String args[] ) {
                System.out.println( "Hello World!" );
        }
}
```

Displaying Hello World to the screen in JavaScript:

```
console.log('Hello World');
```

One other advantage of learning JavaScript is that, since it is the language of the Web, you would be able to share your creations with other people in a really easy manner. I think to be able to do so and receive feedback is an important consideration when learning a new skill set.

To summarize, there are lots of reasons to learn programming and JavaScript stands to be a great choice since it:

- is easier to write;
- is popular and ubiquitous;
- has a vast application domain.

Why Do We Have Different Languages?

You must be wondering why there are different languages if they are all share similar features.

That's a great question. Different languages exist because they are designed with different principles in mind. Some of them can be harder to type out, but they give you more control over the stability and speed of

your programs. Others can be much more concise but could be slower to execute. Some languages are better suited for certain tasks. JavaScript is perfect for full stack web development, Matlab is great for mathematical calculations, C++ has dominance in game programming, and Julia is used for data science. This doesn't mean you can't be using other languages in these domains, though. Unity Game Engine offers C# for game development. Python can be preferred for data science. And GoLang or many other languages could be used for back-end web development. It sometimes boils down to what the developers prefer to use and what they already know. And sometimes it comes down to the constraints of a given project.

I used to work in the visual effects industry and the software packages that we would be using in the field could be automated using Python or C++. So those were great language choices for that domain given that's what the tools that we were using were supporting. Knowing Java in visual effects would have been largely useless except for the fact that knowing a programming language actually makes it more likely that you will be able to pick up another language as they share similar principles among each other.

Choosing which language to learn as your first can sometimes be a tough choice as there are lots of viable options out there. Sometimes the choice is dictated by the application domain. If you are seeking to build a game using the Unreal Engine maybe you should just learn C++. But then again if it's your first time interfacing with a programming language, you might be faced with such a steep learning curve that it might be discouraging.

JavaScript stands to be a great choice to learn as your first programming language. As mentioned earlier, it is widely used and has a vast application domain that would allow you to experiment with lots of different applications. It has a big and active community behind it and also has a very concise syntax that makes it closer to human languages.

Learning JavaScript with p5.js

One of the most challenging aspects of learning programming is to find engaging examples that are not only fun and impressive but also illustrative of the subject matter at hand. Once you get the hang of it, programming is a highly rewarding and engaging activity, but to a beginner most of the problems that a professional programmer has to tackle might seem uninteresting or straight up boring. That's why this book uses a JavaScript library, an add-on, called p5.js in teaching this introduction to programming book. p5.js will allow you to create engaging interactive and visual pieces that you will have fun while creating, and it will also let you build a strong foundation for software development. The visual nature of this library will allow us to actually see the results from our scripts as graphics and develop an intimate understanding of programming structures.

p5.js is a programming library. A programming library can be thought as a collection of code that is built for a specific purpose, so that whenever you need to perform an action that relates to that purpose you can use a library instead of building that functionality yourself. Libraries build on and extend the core capabilities of a language. For JavaScript, there are more than a hundred thousand libraries out there that allow you to perform a large variety of operations. So it is always a good idea to check if someone already created an open source or even a paid library for your needs before trying to implement your own functionality. The idea is that a library would be a battle-tested solution for a particular problem that you can utilize with confidence, instead of devising your own solution, which might introduce unforeseen problems into the program you are developing. This is particularly true for JavaScript as the core language doesn't have any built-in, standard, library; and hence development efforts

rely heavily on external libraries to tackle common problems. Here are examples for a couple of interesting libraries to give you a taste of what is available out there:

- Faker.js (https://github.com/Marak/Faker.js): Generate massive amounts of fake data.
- franc (https://github.com/wooorm/franc): Detect the language of a given text.
- jimp (https://github.com/oliver-moran/jimp): An image processing library.
- cylon.js (https://cylonjs.com/): A robotics framework for robotics, physical computing and the Internet of Things.

p5.js is a creative coding library that is based on the idea of sketching. Just like how sketching can be thought of as a minimal approach to drawing to quickly prototype an idea, p5.js is built on the concept of writing the minimal amount of code to translate your visual, interaction, or animation ideas to the screen. p5.js is a JavaScript implementation of the popular library called Processing, which is based on the Java programming language.

It is worth mentioning that Java and JavaScript are completely unrelated languages. The reason why JavaScript is named after Java is an unfortunate branding and marketing decision made back in the day.

The concise nature of p5.js makes it a very easy library to learn. But don't let this simplicity trick you into believing that p5.js has limited capabilities. p5.js has an impressive amount of functionality, history, and community behind it to make it a valuable learning investment if you ever wanted to create art, design, motion, or interactive pieces using code. A p5.js program can be anywhere from a few lines of code to

thousands. Since p5.js was built with simplicity in mind, sometimes small p5.js programs are referred to as sketches. Even though that's a clever way to describe it, I am personally not a huge fan of that wording since it obfuscates the fact that what you are doing is programming after all.

You can find pragmatic applications of p5.js such as creating data visualizations (Figure 1-1).

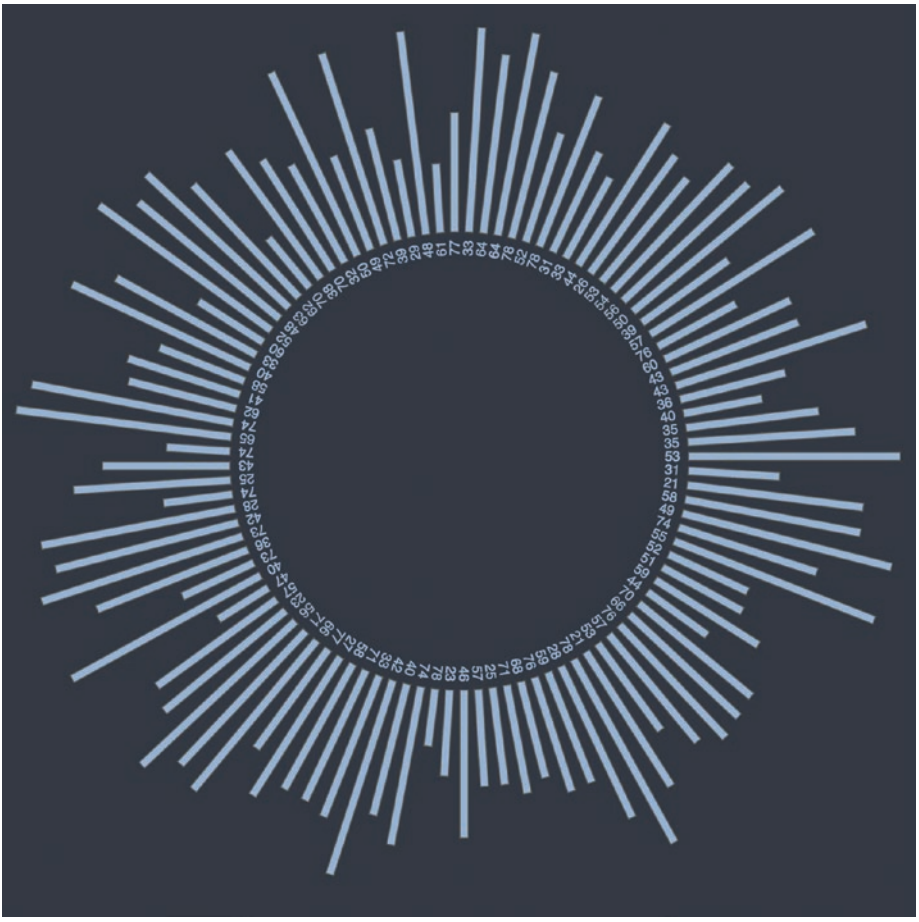

Figure 1-1. *Data visualization with p5.js*

Or you can use it to create abstract generative art (Figure 1-2).

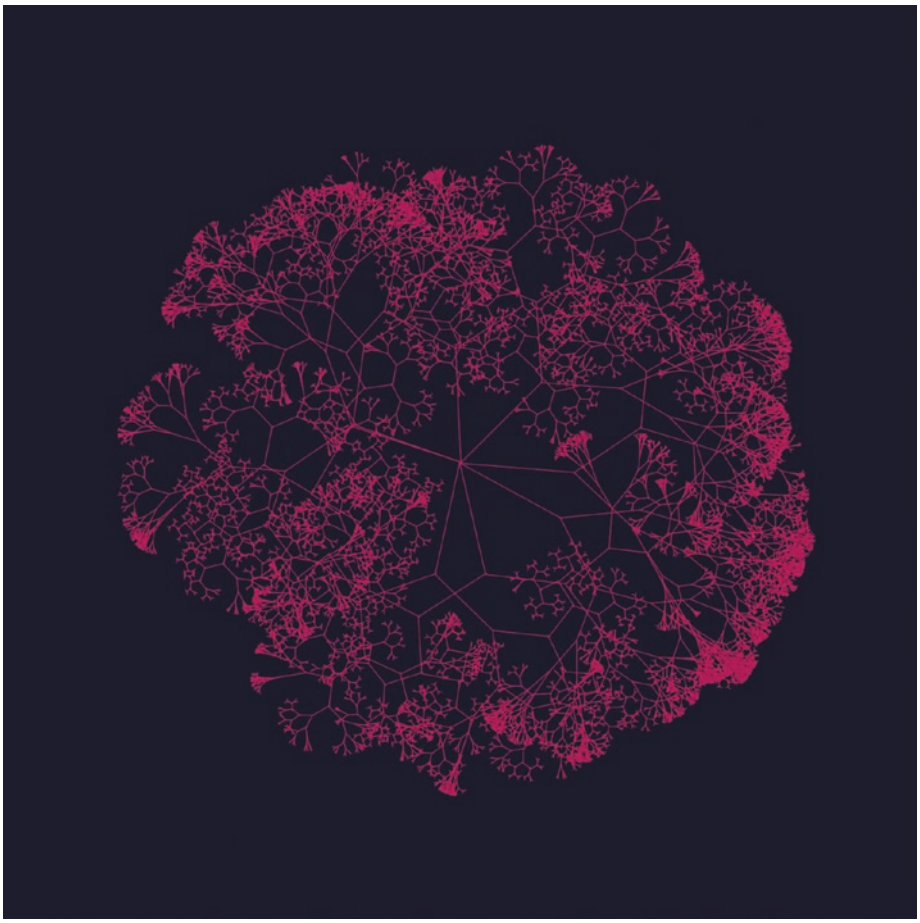

Figure 1-2. *Abstract generative art with p5.js*

You can even create animated or interactive visuals. We will be building an interactive game by the end of this book using p5.js!

CHAPTER 2

Getting Started

Installing p5.js

There are a couple of ways you can start using p5.js and JavaScript. One option is to visit the p5.js website (`https://p5js.org/download`) and download the p5.js source code on to your system (see Figure 2-1).

At the time of the writing of this walkthrough, the download page has a link called 'p5.js complete' that includes the p5.js library and an example project. Download this archive file and find the folder named `empty-example` inside it. In this folder, you will find two files: the `sketch.js` file where you can write JavaScript code and an `index.html` file that can be launched with a web browser such as Chrome and would execute and display the result of the JavaScript code inside `sketch.js` file. You can also find a copy of these files on my GitHub repository: `https://github.com/hibernationTheory/p5js-complete`.

Even though you can change the contents of `sketch.js` JavaScript file with a plain text editor like *NotePad*, you might instead want to use a code editor such as `Sublime Text` to do so.

A code editor is pretty similar to a text editor, like *Notepad* or *Word*, but it has special features that make coding much easier such as highlighting of special words for a given programming language, which in this case that language is JavaScript. Sublime Text is a code editor that you can use, which can be downloaded and evaluated for free.

E. Arslan, *Learn JavaScript with p5.js*, https://doi.org/10.1007/978-1-4842-3426-6_2

Perhaps the easiest way to get started with p5.js is to use an online editor. An online code editor can be used inside the web browser and doesn't require you to install anything on your system. It is my preferred way of working when I am learning as it makes it really easy to get started.

An easy-to-use Online Code Editor that is available at the time of the preparation of this book can be found at this link:

[p5.js online editor - alpha](https://alpha.editor.p5js.org/)

If the above link is not accessible for any reason, you can also try the p5.js template that is hosted on my Codepen account:

[Codepen - p5.js simple template](https://codepen.io/enginarslan/pen/qpBBXz?editors=0011). CodePen (https://codepen.io) is a social development platform that allows you to write code in the browser and share your work with other developers. It is a great environment for development and experimentation. The difference between Codepen and the p5.js editor that is mentioned above is that the p5.js editor only allows you to run p5.js related code inside itself whereas Codepen can execute any front-end code.