

01 - Introduction

Introduction to the Course

WE'LL COVER THE FOLLOWING ^

- ABOUT THIS COURSE
 - Why Learn Coding?
 - Coding vs Programming
 - On HTML and CSS
 - Why Learn JavaScript?
 - Why Do We Have Different Languages?
 - Learning JavaScript with p5.js
 - About This Course

ABOUT THIS COURSE

Learning how to code can be challenging. It is hard to find good resources that are relevant, practical and engaging at the same time. This book teaches coding from scratch using the vastly popular programming language JavaScript. Using JavaScript and a library called p5.js, you will learn how to code in a highly engaging and visual manner.

We will see what p5.js is later in this chapter. At this point it suffices to say that it allows you to create all sorts of visuals such as building sophisticated data visualizations:

Output

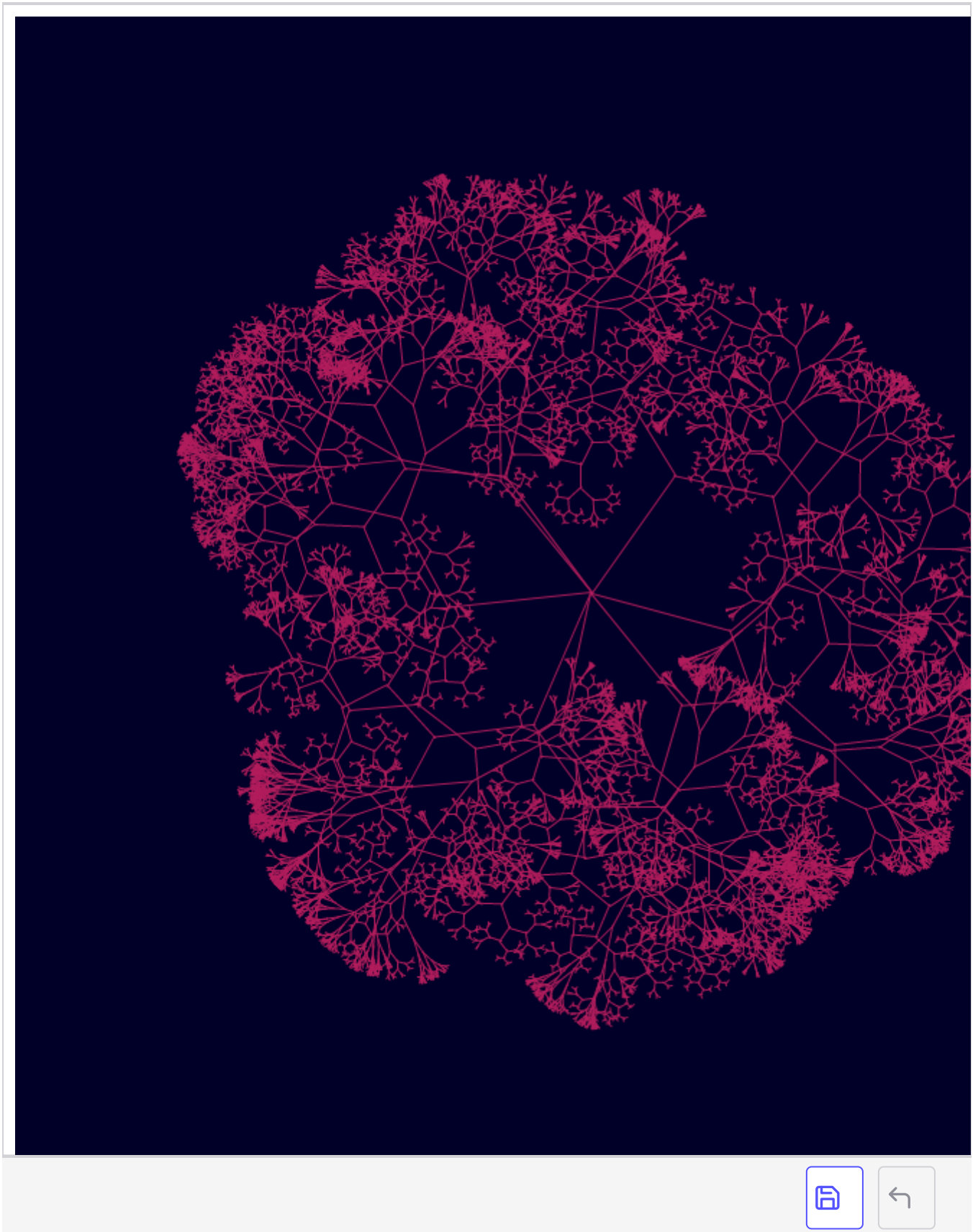
JavaScript

HTML



or to create computational and generative art:

Output
JavaScript
HTML



You can even create animated or interactive visuals:

Output
JavaScript
HTML



with

We will be building an interactive, animated game that can be deployed online at the end of this course!

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 that 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 makes 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 so have been immensely gratifying. Not only it allowed me to become more accomplished in my work in Visual Effects and create award winning effects,

but also empowered me to transition to an even more rewarding career in Software Development.

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.

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 general population. Basically the premise is that you could be coding and still be contributing to 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 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 truth be told these two terms are used almost interchangeably nowadays. The main purpose of this course 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. So HTML teaches you how to write text for 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 allows the computer to perform *logical* operations. Some of the examples to such structures, that we will see more about later, are: conditionals which allows the program to output different results based on given conditions, 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 course is for us to learn about all these fundamental programming concepts.

Almost all programming languages have these kind 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, 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 makes people miss out on what could be achieved when using programming languages.

In this course 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 course 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 uses 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 - backend - 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 someone else also had the same problem, posed a question on this website and got an answer which 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. Here are some examples where a simple statement that displays the words 'hello world' to

the screen is written by using different languages. Notice how much shorter it is to write the same code using JavaScript.

Displaying Hello World to the screen in C++ (Source: <http://helloworldcollection.de/>)

```
#include <iostream>
using namespace std;

int main() {
    // your code goes here
    cout << "Hello World";
    return 0;
}
```



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
- 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 sharing 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 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 a dominance in game programming. **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 backend 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 Visual Effect 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 was 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 which would allow you to experiment with lots of different applications. It has a big and active community behind it and have a very concise syntax which 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 an 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 course uses a JavaScript library, an add-on, called **p5.js** in teaching this introduction to programming course. p5.js will allow you to create engaging interactive and visual pieces that you would 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 hundred thousands of libraries out there that allows 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 couple of interesting libraries to give you a taste what is available out there:

Faker is (<https://github.com/Marak/Faker.js>) : Generate massive amounts

- `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.

About This Course

The emphasis of this course 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 course are highly transferrable and could be used whatever you choose to build; whether web applications, programmable robots or generative art. This means that I will provide you with enough

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 course that I wished I had available when I was learning coding myself.

If you are an artist or a visual designer this course 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 course for learning programming as the visual nature of the exercises will help you grasp the fundamentals of programming easier and let you build a strong foundation in a shorter amount of time.

This course 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 course!

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

- **02 - Getting Started** We will learn some very basic JavaScript commands and operations to get started with using p5.js
- **03 - Colors in p5.js** This will be a p5.js specific chapter where we learn about how colors 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.
- **04 - Operations and Variables** We will make use of the JavaScript knowledge we acquired in the second chapter in p5.js context.
- **05 - If - Else Statements and Comparison Operators** This chapter will allow us to write programs that can respond to different conditions by using conditionals and comparison operators.
- **06 - More p5.js Variables** This will be another p5.js specific chapter where we will learn about several library specific variables.
- **07 - Loops** Here we will learn about loops which will allow us to build programs that handles enormous amount of calculation

- **08 - Functions** Functions are the building blocks of JavaScript and we will learn more about them to build more scalable, modular and robust programs in this chapter.
- **09 - Objects and 10 - Arrays** Objects and Arrays are JavaScript data structures that will allow us to organize our code and handle complexity in more intelligent ways.
- **11 - Events** p5.js Event handling will allow us to write programs that handles user interaction.
- **12 - More on p5.js** Another p5.js only chapter where we learn more about library specific features before diving into our final project.
- **13 - Main Project** We will build a game using everything we have learned up to this chapter!