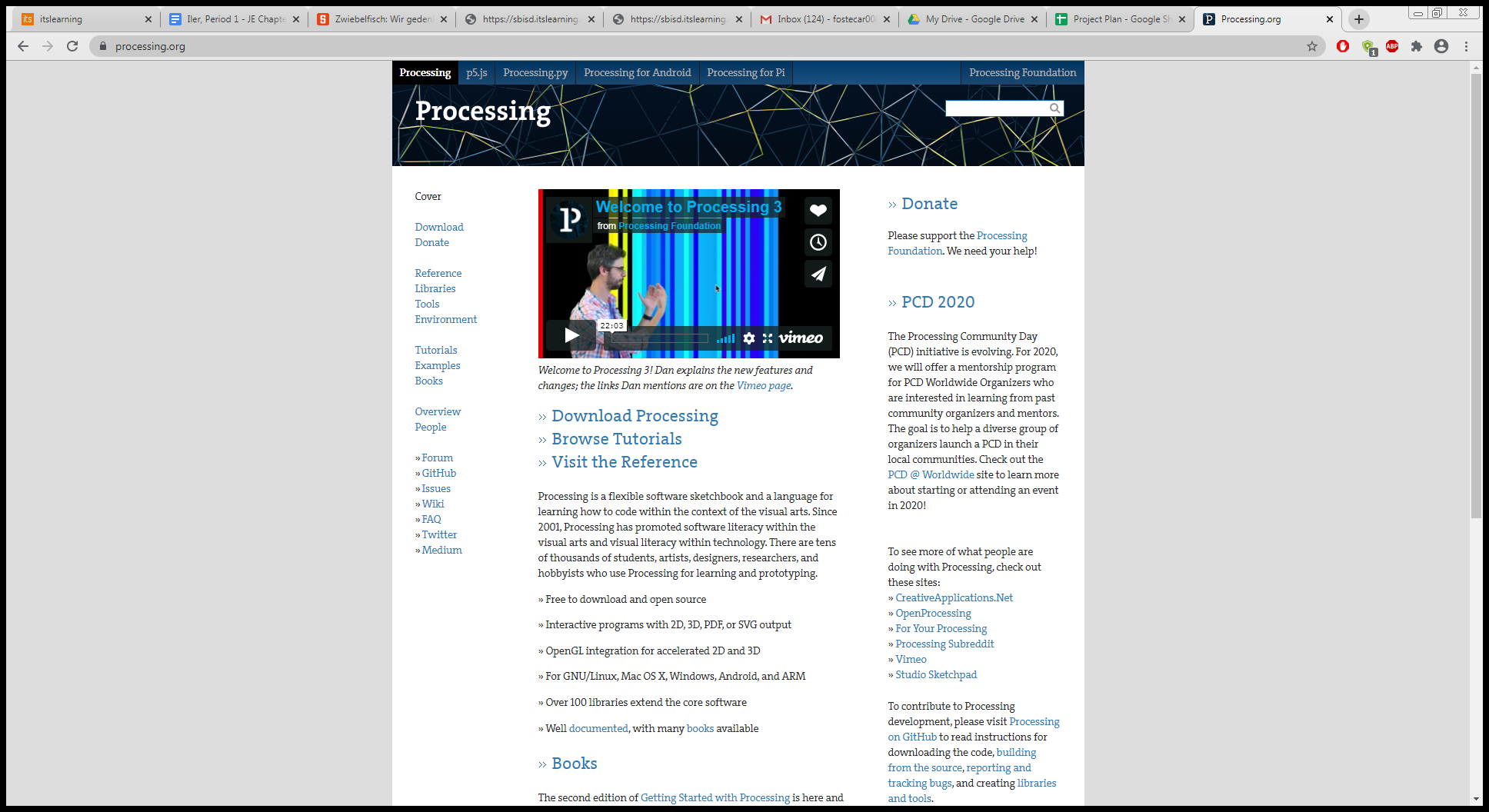
**Installation and Introduction to Processing**

**Carson Foster**

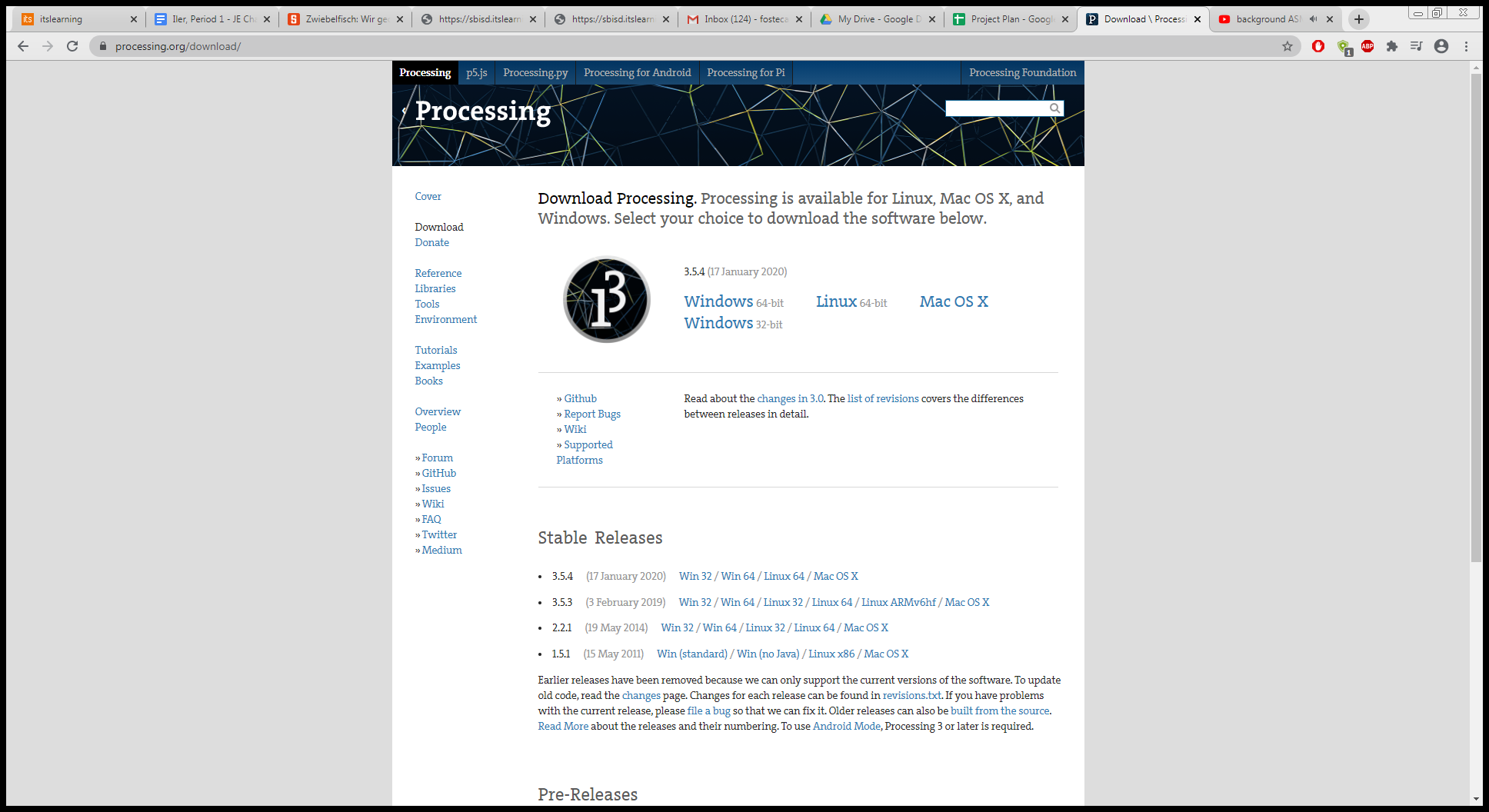
This lesson will cover both an introduction to Processing and how to install the development environment that we will be using.

**Installation**

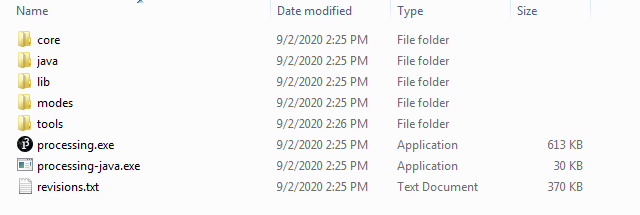
1. Visit <https://processing.org/>
2. Click on “Download Processing”



1. Download the correct version of Processing for your machine.



1. On your machine, extract the archive file you downloaded in step 3. On Windows and Mac, the archive is a .zip file. For Linux, the archive is a .tar.gz file.



1. That’s it. Processing has been ‘installed’ to your machine! There is no formal installer, so just run the executable you extracted to start the Processing Development Environment.

**Introduction**

Processing is designed to be a simple-to-use language that focuses on graphics and animation. At first, it was just an extension to Java, but now it has differentiated itself from its former parent language. Despite this, it is very similar to Java and can be integrated with Java libraries and files.

This series is designed for programmers with Java knowledge already. I’ll focus on Processing-specific methods and properties, and I’ll note differences from Java when applicable.

Alright, enough introduction, let’s write your first Processing program!



Boom, that’s it. Seem familiar? That’s because it’s the main code of the first Java program you wrote. With Processing, you don’t need to write any of the “boilerplate” code before you can hop into the main section of code. This program prints the string “Hello world” to the console. In fact, with Processing, we can condense this program even more, like so:



This does the exact same thing as the above program, but with fewer characters. Now, we’ll look at a program that has a similar effect, but is much longer. Let’s hop in! In case you want to copy and paste code from any of my code sections, double click on a code section to open the embedded word document. Then you’ll be able to select and copy the code.

This program also tells the world hello, but in a slightly different manner. Notice the unfamiliar method calls: these are from the Processing library. If you weren’t sure, the line method creates a 2D line on the screen. We’ll discuss the other functions in this code snippet later in the series, but you can make a reasonable guess at what they do just from reading the code. Once again, note that there is no class or main method definition necessary to run the program. Finally, observe how the syntax is essentially the same as Java’s, with my for loop looking exactly like a Java for loop.

After you finish reading through the code on the next page, take a look through the assignment document and attempt the assignment. I hope that you enjoy your journey through Processing with me!



*The output of the program to the left*