

Creating a Local p5.js Project

Learn how to set up p5.js projects from scratch on your own computer.

When you are ready to move off the Codecademy platform and begin creating your own p5.js projects from scratch, there are a few options available for you to try. The first is with the [p5.js web editor](#), an online code editor for p5.js projects.

The other option is to create your p5.js projects locally on your computer. Creating p5.js projects locally will allow you to save your p5.js sketches for easy access without worrying about the size of your media assets, as you may have to with the web editor. You will also be able to run p5.js sketches without an internet connection.

In this article, you'll learn how to set up files for your local p5.js project, run projects directly on your personal computer, and view the output on a web browser.

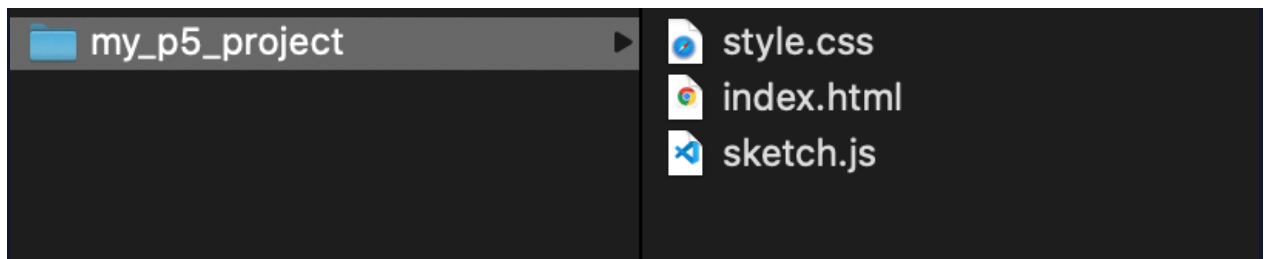
Step 1. Setting up Your Code Editor

Before getting into creating files for your local p5.js project, make sure you have a code editor set up on your computer. If you haven't used a code editor before, take a look at the [Sublime Text Editor](#) article, which goes over the process of choosing and setting up a code editor.

Step 2. Creating the Files

In [Introduction to Creative Coding with p5.js](#) lesson, you learned that a typical p5.js project consists of **index.html** for the web page content including the `<canvas>` HTML element, **style.css** for styling the web page, and **sketch.js** for the JavaScript code for your p5.js sketch.

Create a folder to contain the files for your p5.js project. Inside the folder, create **index.html**, **style.css**, and **sketch.js** files. You can create these files by opening your code editor, creating a new file, and saving it as the name of each of the three files, with the right extensions. In the image below, we've created a folder called **my_p5_project** and the required files inside the folder.



Step 3. Downloading the p5.js Library

To start programming a local p5.js project, the p5.js library files need to be downloaded and added to the folder you created in the last step. There are a few different ways to do this.

The first method is to download the complete library from the [p5.js website](#). This download includes the p5.js library file, the [p5.sound library](#) add-on file, and an example project.

Complete Library

This is a download containing the p5.js library file, the p5.sound addon, and an example project. It does not contain an editor. Visit [Get Started](#) to learn how to setup a p5.js project.

p5.js complete

Includes:

p5.js, p5.sound.js, and an example project

Version 1.1.9 (July 22, 2020)

Another method is to download a single library file. There are two types of files available: **p5.js** and **p5.min.js**. The main difference between these two files is that **p5.min.js** is a compressed version of the **p5.js** file, meaning it is much smaller in size, which enhances load speed when the project is deployed on the web.

Single Files

These are downloads or links to the p5.js library file. No additional contents are included.

p5.js

Single file:
Full uncompressed version

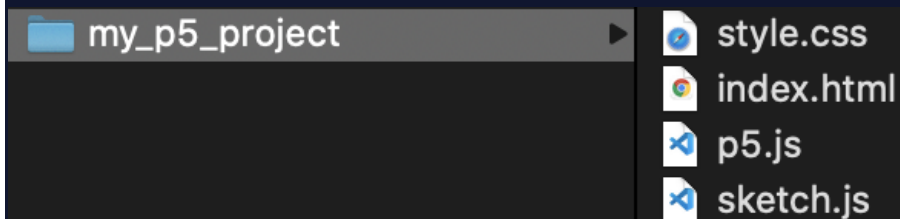
p5.min.js

Single file:
Compressed version

CDN

Link:
Statically hosted file

After downloading the preferred file, your directory should now look something like this:



As an alternative to downloading the library file, you can link to a p5.js file hosted online at a Content Delivery Network. A Content Delivery Network (CDN) is a network of servers that allows content to be distributed and delivered to users. By choosing to link to a p5.js library file hosted at the p5.js CDN, you are receiving the file from a remote server.

The [p5.js CDN page](#) has links to the recent versions of the p5.js library files. You can either copy the URL link to use as the `src` property of the `<script>` HTML tag or copy the entire `<script>` tag from the CDN website to include the library inside the `<head>` HTML element of **index.html** like below:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.2.0/p5.min.js"></script>
```

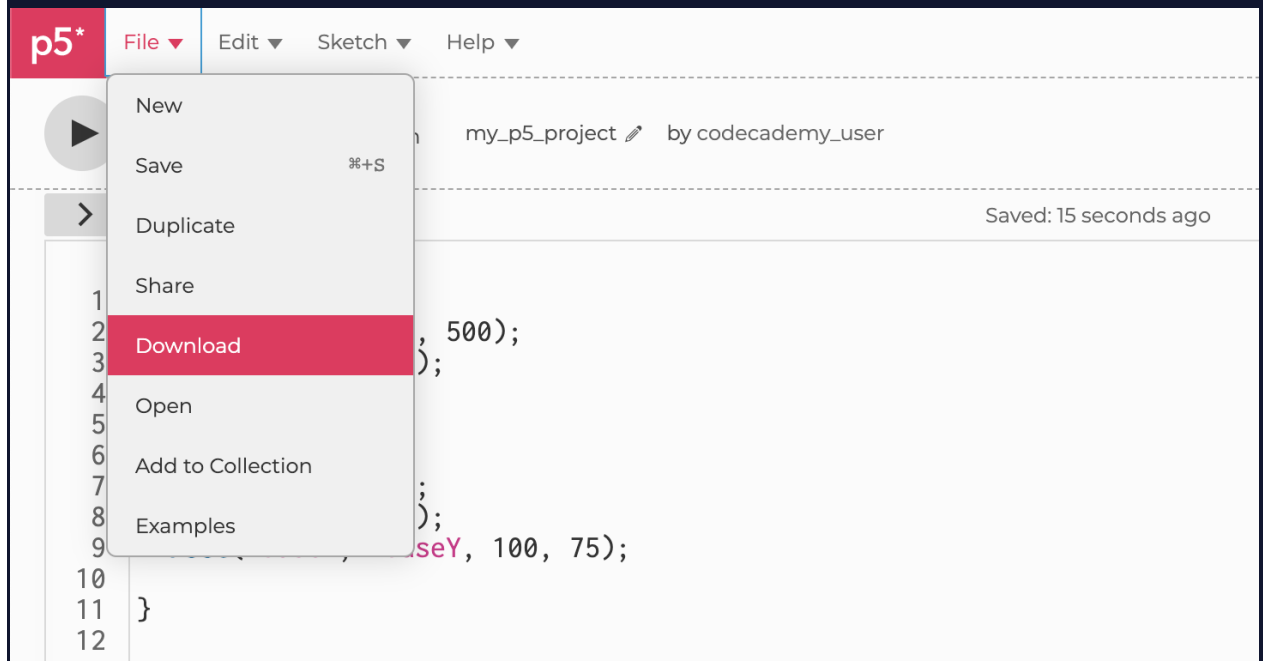
Don't worry if you don't know where this `<script>` tag is supposed to go. We will talk more about it later in the article.

[Optional Step] Downloading Your Project From p5.js Web Editor

If you have a project that you've been working on in the p5.js web editor and decide that you would like to continue working on it locally, you can do so by

downloading your files. Navigate to the File drop-down menu in your project's p5.js web editor page and click on the "Download" button. You should see a .zip file downloaded to your computer with all of the required files for running the project locally.

Unzip the downloaded file, and the complete p5.js directory structure will be available for you to access locally.



Step 4. Setting up index.html

Just like a normal Javascript program, to run any p5.js code, the **sketch.js** file, **style.css** file and all downloaded library files like **p5.js** need to be added into the **index.html** file.

Inside the `<head>` element of the HTML file, add the **style.css** file using a `<link>` tag and add the **p5.js** (or **p5.min.js**) file using the `<script>` tag. The **sketch.js** file can be added into the `<body>` element of **index.html** using a `<script>` tag as well. The **sketch.js** file is typically added to the `<body>` element because sketch.js needs to run after the p5.js library is fully loaded and ready to be used.

Your HTML file should look like this:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script src="p5.js"></script>
    <meta charset="utf-8">
  </head>
```

```
<body>
  <script src="sketch.js"></script>
</body>
</html>
```

Using your preferred code editor, feel free to add this exact HTML code into your **index.html** file. If you decide to add the code above into your HTML file, make sure that the names of all your files match with the ones copied over.

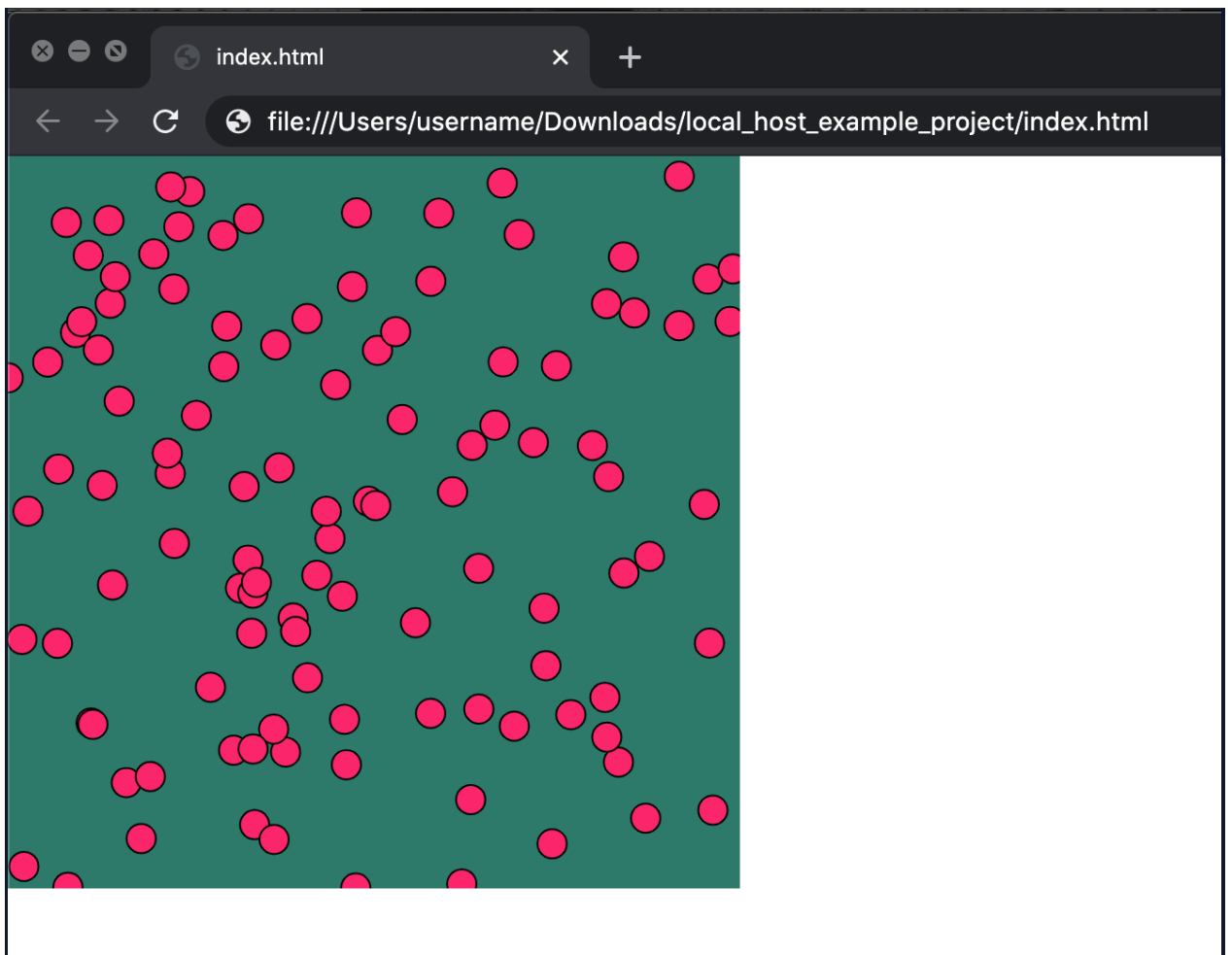
Step 5. Adding Styles to style.css

It is also important to make sure that the **style.css** file is properly set up. The CSS code below adds some basic styling to the p5.js sketch. Feel free to copy and paste it into the **style.css** file in your local directory.

```
html, body {
  margin: 0;
  padding: 0;
}
canvas {
  display: block;
}
```

Step 6. Running Projects Without Media Assets

If your program does not contain any media assets, such as images, videos, and sound files, then you can run your local p5.js program simply by clicking on the **index.html** file in your project directory. This will cause your default browser to open, render, and show the output of your code from your project's local file path. The screenshot below gives you an example of what the file path could look like when you run a local p5.js project without any media files.



Steps 7 and 8 do not apply to projects without media assets, so you can move directly to step 9 to learn about how to debug your p5.js code. However, if your project does contain media assets, then opening the **index.html** file will not render your sketch. To learn how to set up a local server to view your projects with media assets, continue to step 8.

In a later p5.js lesson, you will learn how to add media assets to sketches. After you go over the [Media](#) lesson, you can come back to this article to learn how to run p5.js projects with images and videos locally.

Step 7. Creating a Local Server

If your project uses media assets, it is necessary to set up a local server to avoid Cross-Origin Resource Sharing (CORS) errors when loading the webpage. [CORS](#) is a mechanism put in place to ensure web security standards are met when requests for loading resources are made.

Local servers create a way for testing code without having to deploy it to an external web server. Essentially, they are servers deployed onto a local

machine, which you can access as a local user. For more information about local servers, this [p5.js wiki article](#) written by the processing foundation is a helpful resource.

There are a few ways of setting up a local server using Python 2 and Python 3. Each method requires using the terminal if you are on Mac or Linux or using the command prompt on Windows. If you are unfamiliar with using the terminal/command prompt, review this [Command Line Interface Setup](#) article. The article also contains information about navigating directories using terminal commands, which will be necessary to move into your project directory.

Begin by opening up your computer's terminal or command prompt, then use the command line to navigate into the directory in which your p5.js project is located. For example, if the project's path is **User/Desktop/p5_projects/my_p5_project**, you would want to navigate to the **my_p5_project** directory using the `cd` command. Once inside the directory, run the command that matches the version of Python installed on your computer.

If you have Python 2 installed on your computer, then you could use this command:

```
python -m SimpleHTTPServer
```

If you have Python 3 installed on your computer, then you could use this command:

```
python3 -m http.server
```

To check which version of Python is running on your computer, use this command:

```
python -V
```

If you are unfamiliar with Python packages and whether or not you have Python installed on your computer, check out this [Installing Python 3 and Python Packages](#) article.

Step 8. Viewing the Locally Hosted Project In the Browser

You are almost done with the setup! All that is left is opening your browser and typing either

`http://localhost:xxxx`

where `xxxx` is the specified port number, or,

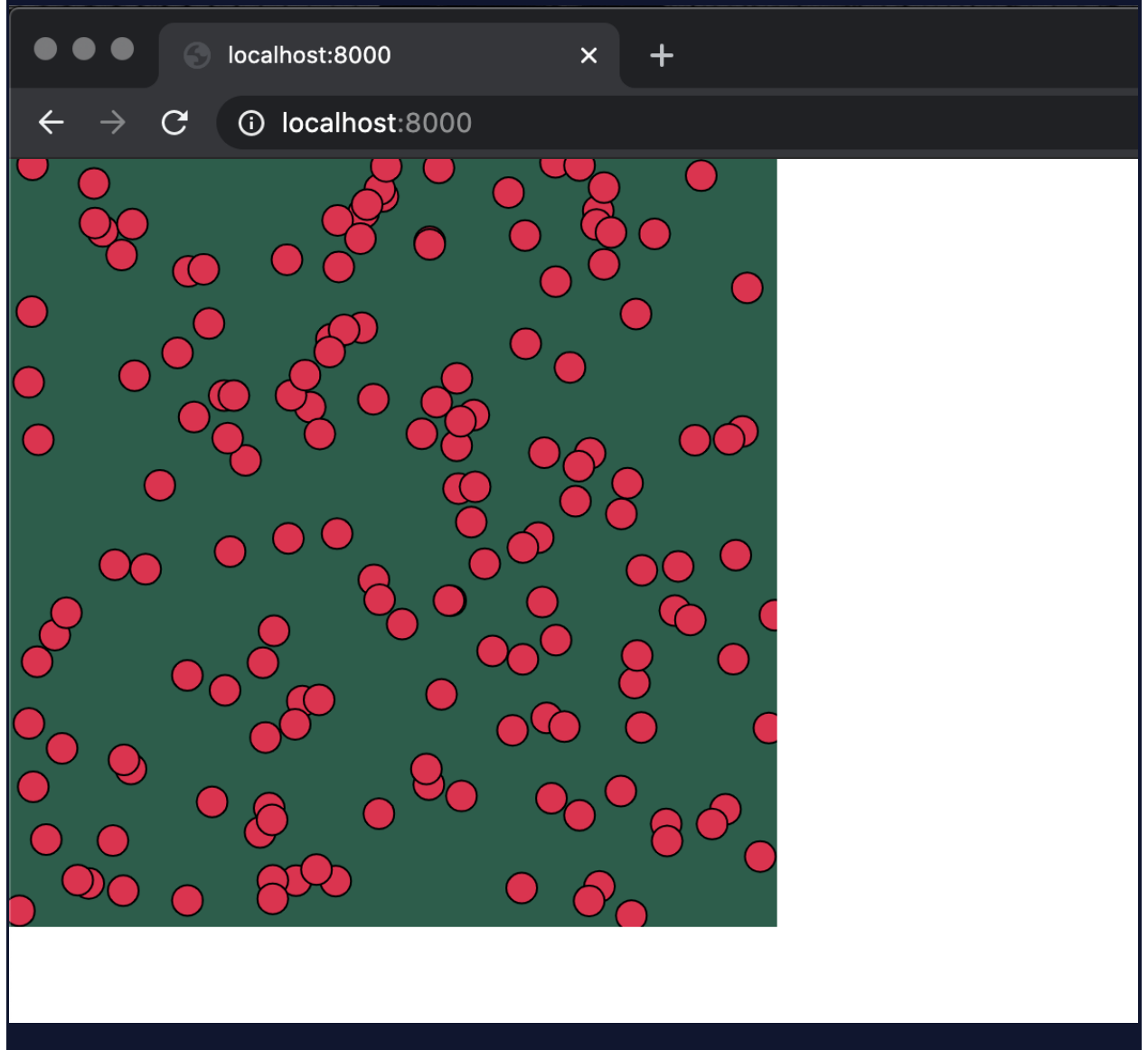
`http://127.0.0.1:xxxx/`

where `xxxx` is the specified port number.

If you used one of the commands in step 7, then the default port will be 8000. If you are unsure what port number your local server is running on, go back to your terminal to find out by reading the lines after the local server command was called.

Serving HTTP on :: port 8000 (http://[::]:8000/) ...

The screenshot below shows an example p5.js sketch running on localhost on port 8000.



Once you enter either of the previous localhost addresses into the browser's URL, you should now be able to see the p5.js sketch in the browser! If you still do not see the p5.js sketch, double-check that the local server was created in the same directory as your p5.js project.

Step 9. Using the Browser's Console

It is important to remember that p5.js is a Javascript framework and as such, a p5.js project runs like any other web project. This means that if you want to debug your code or log values to the console, you can use the browser's development tools. The [Use DevTools](#) article dives into development tools in extensive detail if you want to learn about how to debug your web projects.

Review

Over the course of this article, you learned how to create p5.js projects on your computer and run them locally! You also learned about the terminal, how to host a local server, and how to properly set up files for local p5.js development projects. You may want to revisit this article as you make your progress through the [Learn p5.js course](#) or if you would like a refresher on how to create local p5.js projects.