## **Etch-A-Sketch**

### **Introduction**

In this project, you’ll be creating a pretty neat toy for your portfolio to flex your DOM manipulation skills. You’re going to build a browser version of something between a sketchpad and an Etch-A-Sketch.

This project should *not* be easy for you. You’ll probably have to Google frequently to get the right JavaScript methods and CSS to use – in fact, that’s the point! You *can* build this using the tools that you have already learned, and there are plenty of resources on the net for learning stuff that we haven’t covered yet if you decide you need it. We’ll walk you through the basic steps, but it will be up to you to actually implement them.

If you get totally stuck, drop by the chatroom. Someone will be there to point you in the right direction.

### **Assignment**

1. [Follow the instructions atop Odin’s Recipes project](https://www.theodinproject.com/paths/foundations/courses/foundations/lessons/recipes#setting-up-your-projects-github-repository) to set up a git repository for this project.
2. Create a webpage with a 16x16 grid of square divs.
   1. Create the divs using JavaScript. Don’t try making them by hand with copy and pasting in your html file!
   2. It’s best to put your grid squares inside another “container” div (which can go directly in your html).
   3. There are several different ways to make the divs appear as a grid (versus just one on each line). Feel free to use any or play with each of them:
      1. float/clear
      2. inline-block
      3. flexbox
      4. CSS Grid
   4. Be careful with borders and margins, as they can adjust the size of the squares!
   5. “OMG, why isn’t my grid being created???”
      1. Did you link your CSS stylesheet?
      2. Open your browser’s developer tools.
      3. Check if there are any errors in the JavaScript console.
      4. Check your “elements” pane to see if the elements have actually shown up but are somehow hidden.
      5. Go willy-nilly and add console.log statements in your JavaScript to see if it’s actually being loaded.
3. Set up a “hover” effect so that the grid divs change color when your mouse passes over them, leaving a (pixelated) trail through your grid like a pen would.
   1. Hint: “Hovering” is what happens when your mouse enters a div and ends when your mouse leaves it. You can set up event listeners for either of those events as a starting point.
   2. There are multiple ways to change the color of the divs, including:
      1. adding a new class to the div.
      2. changing the div’s background color using JavaScript.
4. Add a button to the top of the screen which will clear the current grid and send the user a popup asking for the number of squares per side for the new grid. Once entered, the new grid should be generated *in the same total space as before* (e.g. 960px wide) so that you’ve got a new sketch pad. Tip: Set the limit for the user input to a maximum of 100. A larger number of squares results in more computer resources being used, potentially causing delays, freezing, or crashing that we want to prevent.
   1. Research button tags in HTML and how you can make a JavaScript function run when one is clicked.
   2. Also check out prompts.
   3. You should be able to enter 64 and have a brand new 64x64 grid pop up without changing the total amount of pixels used.
5. (Optional): Instead of just changing the color of a square from black to white (for example), have each pass through with the mouse change it to a completely random RGB value. Then try having each pass just add another 10% of black to it so that only after 10 passes is the square completely black.
6. Push your project to GitHub!

Solutions

<https://michalosman.github.io/etch-a-sketch/>

<https://rlmoser99.github.io/etch-a-sketch/>

<https://bscottnz.github.io/esketch/>

<https://andrewjh271.github.io/etch-a-sketch/>

<https://artis-dev.github.io/etch-a-sketch/>