My Little Rainbow

[(https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab)](https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab) [(https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab/issues/new/choose)](https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab/issues/new/choose)

Introduction

In this lab, you're going to make a rainbow with HTML `<div>` elements. Through this process, you'll learn about HTML elements, CSS styling, CSS selectors, how color works in CSS and importing stylesheets. That might sound like a lot but it isn't.

Before you start, here's some basic info about HTML and CSS. Skip to **Making a Rainbow** if you feel comfortable enough with HTML and CSS.

Objectives

You will be able to:

- Describe the purpose of IDs in HTML
- Add ID elements to HTML tags
- Select and style HTML elements with CSS

HTML Basics

Hyper Text Markup Language, or HTML, is a way to demarcate a document into different parts. Each part is *marked* by elements (using tags). Each element has its own special connotation that the browser uses to *render* the HTML document. Use this **cheat sheet (https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf)** on HTML elements for guidance.

Elements

- All begin with `<` and end with `>` , e.g., `<div>` (this last part is a tag).
- Most have an opening tag such as `<div>` and a closing tag `</div>` .
  - The `/` indicates to the browser that that tag is a closing tag.
  - The element is everything between the tags and the tags themselves.
- Some tags are self-closing like the line break element `<br>` .
- Elements can have IDs and classes to aid the browser in finding specific tags.
  - Must begin with a letter A-Z or a-z.
  - Can be followed by: letters (`A-Za-z`), digits (`0-9`), hyphens (`-`), and underscores (`_`).
  - IDs **can** only be used once per page, e.g.: `<div id="this-special-div"></div>` .

    - Classes can be used as many times as you want, e.g.: `<div class="a-less-special-div"></div>` .
- Elements nested inside other elements are called children.
    - Children inherit attributes from their parents.
    - Don't nest everything.
- Elements next to one another are siblings.
    - Siblings do not inherit from one another but are important for selecting in CSS.

Here is an example of element relations: `html <div> <!-- the parent element --> <p></p> <!-- the first sibling element/the first child--> <p></p> <!-- the second sibling element/the second child --> </div>`

CSS Basics

Cascading Style Sheets, or CSS, is a language created to style an HTML document by telling the browser how specific elements should look. CSS does this by selecting elements based on their tag, ids, classes, or all of the above. The reason for CSS is the separation of concerns. You want HTML only to be concerned with how it displays and demarcates information, and we let CSS worry about how to make that information look pretty.

CSS selectors

- They select elements to assign them styles.
- `*` (wildcard) selects every element.
- An element, such as `div` , will select all elements of that type.
- They select an id like `#some-id`
- Classes are selected like this `.some-class`
- To select all children elements of a parent do something like this `div p`
- To select multiple different elements separate them by commas like this `div, p, a`

Here's an example of CSS styling:

```
* {
  color: red;  /* color in CSS refers to font color */
}  /* all elements will have red font */
```

Making a Rainbow

First off make sure you have [forked (https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab)](https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab) and cloned this repo. Next, create a new branch, and switch to it; it's `git checkout -b your_solution_branch_name` in case you forgot.

In that directory, you'll see three files. `index.html` , `main.css` , and this `README.md` . Open them in your text editor via your command line. Also, open `index.html` in your browser; if everything is working correctly you should see a white page. Good job!

Making the Divs Visible

If you use the inspector or look at the file in your text editor, you'll see that the basic file structure is there. So why can't we see anything?!?!? That's because the divs have no styling on them right now. And that's because we never told the browser to include a CSS file that would apply any styles.

Let's fix this by adding the stylesheet to the `head` like so, `HTML <head> ... <link rel="stylesheet" type="text/css" href="main.css"> ... </head>` The link is a self-closing tag that will create a relative path with the `href` attribute. A relative path means the browser knows that the `main.css` file is in the same place as `index.html` . The `head` is a hidden part of the page that tells the browser where to find any other files it needs to display the page correctly, the `title` for the tab, and any other possible important information.

Now if you refresh the `index.html` page in your browser you should see an ugly black rainbow. Gerd Jerb!

Okay so you've got the basic outline because in the `main.css` all the `div` elements were selected and styled, but the colors are wrong because the default border color is black. To fix this we need to learn a little bit more about colors, because while we could just set `color: red;` we should learn how to make colors without words. We should just use numbers and not just any set of numbers, but numbers with base pair of 16 rather than base pair of 10 like we use every day. These numbers are called hexadecimal numbers and we can use them to make colors.

Some Stuff You Should Know About Hex Colors (and Web Colors in General).

Hex colors begin with `#` and are followed by, generally, 6 numbers, but some of these numbers are actually letters. The lowest single-digit number in hex is 0 and the highest single-digit number is f. This table might help to visualize what we mean by this.

```
Decimal Numbers:      0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
Hexadecimal Numbers:  0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  a,  b,  c,  d,  e,  f, 10
```

Hex colors work by creating Red, Green, Blue (RGB) values. Traditional RGB colors are on a scale of 0 to 255 for each of the three colors in the spectrum. Hex colors are considered true colors since it can represent ~16 million colors, but your eye can only see 10 million of those. So `#000000` translates to black since 0 reds, 0 green, 0 blues represents the absence of all colors, and `#ffffff` makes white since 255 reds, 255 greens, and 255 blues is the maximum of each of the colors.

Hex colors can be shortened to just three numbers when each RGB value is the same for each digit. So `#11dd99` can be written as `#1d9` .

Coloring the Rainbow

To get roygbiv onto our rainbow we'll need seven hex colors. Red: `#f00` ; Orange: `#ffa500` ; Yellow: `#ff0` ; Green: `#00bc3f` ; Blue: `#06f` ; Indigo: `#8a2be2` ; Violet: `#d300c9`

With those colors, all we have to do next is select each div individually. That is a perfect use for ids since they're meant to style one specific element only. So that means we'll need to add an id for each div, so a logical name for each div would be the color that they have to be. It could be something random, but good names make for semantic code. So let's give the outermost div the id red. We'll do this like so. `html <div id="red"> ... </div>`

And to give that id some CSS attributes we'll go into `main.css` , select the id, and mark its color as red like this.

```
#red { /* this selects any elements with the red id */ border-top-color: #f00; }
```

So to make sure the rainbow isn't so monochromatic you now need to repeat the above steps with the final six colors of the rainbow, and when you do you should have something like **this** **(http://i0.kym-cdn.com/photos/images/original/000/118/087/2468904593_6a7c692ab6.jpg)** .

Summary

Well done! In this lab, you independently modified HTML and CSS files in order to update a webpage. Specifically, you got some experience understanding and working with HTML IDs. You also used CSS to style HTML elements.

How do you feel about this lesson?

👍 👎

Have specific feedback?

Tell us here! **(https://github.com/learn-co-curriculum/dsc-my-little-rainbow-lab/issues/new/choose)**