**Introduction to Color**

CSS supports a wide variety of colors. These include *named colors*, like blue, black, and LimeGreen, along with colors described by a numeric value. Using a numeric system allows us to take advantage of the whole spectrum of colors that browsers support. In this lesson, we’re going to explore all the color options CSS offers.

Colors in CSS can be described in three different ways:

* *Named colors* — English words that describe colors, also called *keyword colors*
* *RGB* — numeric values that describe a mix of red, green, and blue
* *HSL* — numeric values that describe a mix of hue, saturation, and lightness

We’ll learn about and explore the benefits of each of these in depth. Using only named colors, you may feel like you’re picking labeled crayons out of a box. By the end of this lesson, you’ll feel like a painter mixing paints on a palette.

**Instructions**

Proceed to the next exercise when you are ready to begin.

**Foreground vs Background**

Before discussing the specifics of color, it’s important to make two distinctions about color. Color can affect the following design aspects:

1. The foreground color
2. The background color

Foreground color is the color that an element appears in. For example, when a heading is styled to appear green, the *foreground color* of the heading has been styled.

Conversely, when a heading is styled so that its background appears yellow, the *background color* of the heading has been styled

In CSS, these two design aspects can be styled with the following two properties:

1. color - this property styles an element’s foreground color.
2. background-color - this property styles an element’s background color.

h1 {

color: Red;

background-color: Blue;

}

In the example above, the text of the heading will appear in red, and the background of the heading will appear blue.

**Instructions**

**1.**

In **style.css**, set the foreground color of the h1 heading to MidnightBlue.

**2.**

Next, set the background color of the h1 heading to Aqua.

Notice that the background-color property did not change the background color of the entire page. Instead, only the background area behind heading changed. We’ll learn why this occurs in more detail later in the course.

**Note:** The background color of the heading will block the image of the coffee.

h1 {

  color: MidnightBlue;

  background-color: Aqua;

  font-family: 'Covered By Your Grace', sans-serif;

  font-size: 100px;

  line-height: 76px;

  margin: 0;

  position: relative;

  text-align: center;

  top: 20%;

}

# Hexadecimal

One syntax that we can use to specify colors is called hexadecimal. Colors specified using this system are called hex colors. A hex color begins with a hash character (#) which is followed by three or six characters. The characters represent values for red, blue and green.

DarkSeaGreen: #8FBC8F

Sienna: #A0522D

SaddleBrown: #8B4513

Brown: #A52A2A

Black: #000000 or #000

White: #FFFFFF or #FFF

Aqua: #00FFFF or #0FF

In the example above, you may notice that there are both letters and numbers in the values. This is because the hexadecimal number system has 16 digits (0-15) instead of 10 (0-9) like you are used to. To represent 10-15, we use A-F. [Here](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value) is a list of many different colors and their hex values.

Notice that Black, White, and Aqua are all represented with both three characters and six characters. This can be done with hex colors whose number pairs are the same characters. In the example above, Aqua can be represented as #0FF because both of the first two characters are 0 and the second and third pairs of characters are both Fs. Keep in mind that all three character hex colors can be represented with six characters (by repeating each character twice) but the same is not true in reverse.

You can include hex colors just as you would include named colors: background-color: #9932cc;.

**Instructions**

**1.**

In the browser is a web page that uses named colors and hex colors. We’re going to translate the named colors into hex, to be more consistent. The colors won’t visually change, yet.

In **style.css**, find the CSS rule that uses the named color DarkSeaGreen and change it to its hex value, #8FBC8F.

**2.**

Find the four other named colors of the roast types and convert them to their hex values, as shown here:

Sienna: #A0522D

SaddleBrown: #8B4513

Brown: #A52A2A

Black: #000000

# RGB Colors

There is another syntax for representing RGB values that uses decimal numbers. It looks like this:

h1 {

color: rgb(23, 45, 23);

}

Here, each of the three values represents a color component, and each can have a decimal number value from 0 to 255. The first number represents the amount of red, the second is green, and the third is blue. These colors are exactly the same as hex, but with a different syntax and a different number system.

In general, hex and decimal color representations are equivalent. Which you choose is a matter of personal taste. That said, it’s good to choose one and be consistent throughout your CSS, because it’s easier to compare hex to hex and decimal to decimal.

**Instructions**

**1.**

In **style.css**, find the hex value #8FBC8F and change it to rgb(143, 188, 143).

**2.**

In **style.css**, find the hex value #A0522D and change it to rgb(160, 82, 45).

**3.**

In **style.css**, find the hex value #8B4513 and change it to rgb(139, 69, 19).

green {

  background-color: rgb(143, 188, 143);

}

.light {

  background-color: rgb(160, 82, 45);

}

.city {

  background-color: rgb(139, 69, 19);

}

# Hex and RGB

The hexadecimal and RGB color system can represent many more colors than the small set of CSS named colors. We can use this new set of colors to refine our web page’s style.

In both hex and decimal, we have three values, one for each color. Each can be one of 256 values. Specifically, 256 \* 256 \* 256 = 16,777,216. That is the amount of colors we can now represent. Compare that to the 147 named CSS colors!

Recall that we started with named colors, converted them to hex, and then converted some of the hex colors to decimal. Unless we made a mistake, all of the colors should still be the same, visually. Let’s use our broadened palette to make some more refined color choices.

**Instructions**

**1.**

In the .green rule in **style.css**, change the background color to #9EB599.

Run the code. Can you tell the difference?

**2.**

In the .light rule in **style.css**, change the background color to #683C2C.

**3.**

In the .city rule in **style.css**, change the background color to #4C352D.

**4.**

In the .vienna rule in **style.css**, change the background color to #352926.

**5.**

In the .italian rule in **style.css**, change the background color to #141212.

Run the code one more time. These new colors are a lot closer to the real-life color of each type of coffee. How does the subtle difference feel?

.green {

  background-color: #9EB599;

}

.light {

  background-color: #683C2C;

}

.city {

  background-color: #4C352D;

}

.vienna {

  background-color: #352926;

}

.italian {

  background-color: #141212;

}

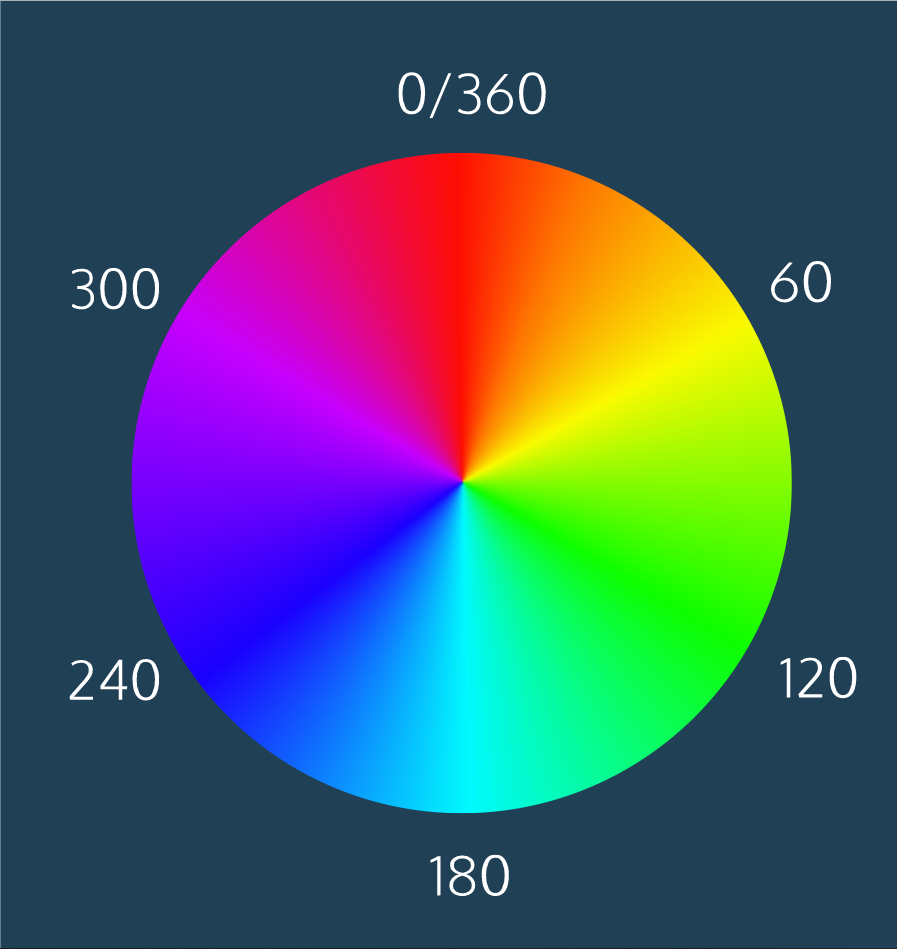
# Hue, Saturation, and Lightness

The RGB color scheme is convenient because it’s very close to how computers represent colors internally. There’s another equally powerful system in CSS called the hue-saturation-lightness color scheme, abbreviated as HSL.

The syntax for HSL is similar to the decimal form of RGB, though it differs in important ways. The first number represents the degree of the hue, and can be between 0 and 360. The second and third numbers are percentages representing saturation and lightness respectively. Here is an example:

color: hsl(120, 60%, 70%);

Hue is the first number. It refers to an angle on a color wheel. Red is 0 degrees, Green is 120 degrees, Blue is 240 degrees, and then back to Red at 360. You can see an example of a color wheel below:



Saturation refers to the intensity or purity of the color. If you imagine a line segment drawn from the center of the color wheel to the perimeter, the saturation is a point on that line segment. If you spin that line segment to different angles, you’ll see how that saturation looks for different hues. The saturation increases towards 100% as the point gets closer to the edge (the color becomes more rich). The saturation decreases towards 0% as the point gets closer to the center (the color becomes more gray).

Lightness refers to how light or dark the color is. Halfway, or 50%, is normal lightness. Imagine a sliding dimmer on a light switch that starts halfway. Sliding the dimmer up towards 100% makes the color lighter, closer to white. Sliding the dimmer down towards 0% makes the color darker, closer to black.

HSL is convenient for adjusting colors. In RGB, making the color a little darker may affect all three color components. In HSL, that’s as easy as changing the lightness value. HSL is also useful for making a set of colors that work well together by selecting various colors that have the same lightness and saturation but different hues.

**Instructions**

**1.**

In the browser is a simple page with different colored rectangles.

In **style.css**, modify the lightness of the background color of the class selector .midground to be 25%.

**2.**

Change the saturation of the background color of the .foreground class selector to 50%.

**3.**

Change the hue of the body selector’s background color to 240 degrees.

body {

  background-color: hsl(240, 100%, 80%);

}

.midground {

  background-color: hsl(225, 100%, 25%);

}

.foreground {

  background-color: hsl(325, 50%, 50%);

}

# Opacity and Alpha

All of the colors we’ve seen so far have been opaque, or non-transparent. When we overlap two opaque elements, nothing from the bottom element shows through the top element. In this exercise, we’ll change the opacity, or the amount of transparency, of some colors so that some or all of the bottom elements are visible through a covering element.

To use opacity in the HSL color scheme, use hsla instead of hsl, and four values instead of three. For example:

color: hsla(34, 100%, 50%, 0.1);

The first three values work the same as hsl. The fourth value (which we have not seen before) is the alpha. This last value is sometimes called the opacity.

Alpha is a decimal number from zero to one. If alpha is zero, the color will be completely transparent. If alpha is one, the color will be opaque. The value for half transparent would be 0.5.

You can think of the alpha value as, “the amount of the background to mix with the foreground”. When a color’s alpha is below one, any color behind it will be blended in. The blending happens for each pixel; no blurring occurs.

The RGB color scheme has a similar syntax for opacity, rgba. Again, the first three values work the same as rgb and the last value is the alpha. Here’s an example:

color: rgba(234, 45, 98, 0.33);

Alpha can only be used with HSL and RGB colors; we cannot add the alpha value to color: green color: #FFFFF.

There is, however, a named color keyword for zero opacity, transparent. It’s equivalent to rgba(0, 0, 0, 0). It’s used like any other color keyword:

color: transparent;

**Instructions**

**1.**

Give the element with class .foreground an alpha value of 0.6. Remember to change hsl to hsla.

**2.**

Give the element with class .midground an alpha value of 0.4 using hsla.

Run the code, and notice how all the colors blend depending on how they overlap.

**3.**

Modify the body selector’s background color to have a value of rgba(0, 255, 0, 0.5).

**4.**

Now, set the alpha of the document body to 0.1. How does opacity change the background?

body {

  background-color: rgba(0, 255, 0, 0.1);

}

.midground {

  background-color: hsla(225, 100%, 25%, 0.4);

}

.foreground {

  background-color: hsla(325, 50%, 50%, 0.6);

}

# Color Review

We’ve completed our extensive tour of the colors in CSS! Let’s review the key information we’ve learned.

There are four ways to represent color in CSS:

* Named colors — there are 147 named colors, which you can review [here](https://msdn.microsoft.com/en-us/library/aa358802(v=vs.85).aspx).
* Hexadecimal or hex colors
  + Hexadecimal is a number system with has sixteen digits, 0 to 9 followed by “A” to “F”.
  + Hex values always begin with # and specify values of red, blue and green using hexademical numbers such as #23F41A.
* RGB
  + RGB colors use the rgb() syntax with one value for red, one value for blue and one value for green.
  + RGB values range from 0 to 255 and look like this: rgb(7, 210, 50).
* HSL
  + HSL stands for hue (the color itself), saturation (the intensity of the color), and lightness (how light or dark a color is).
  + Hue ranges from 0 to 360 and saturation and lightness are both represented as percentages like this: hsl(200, 20%, 50%).
* You can add opacity to color in RGB and HSL by adding a fourth value, a, which is represented as a percentage.

Great job! Feel empowered to add a bit of color to each of your projects!