Codrops

# hsl()

The `hsl()` functional notation is used to set colors in the HSL format, and it takes 3 comma-separated values: hue, saturation, and lightness (or luminance).

HSL stands for "Hue Saturation Lightness".

A hue is specified as an angle within the color wheel (see image below) relative to red. Despite being an <angle>, the value of the angle is unit-less, and defaults to degrees. By definition **red = 0° = 360°**, and the other colors are spread around the circle, so **green = 120°, blue = 240°**, etc. As an angle, it implicitly wraps around such that **-120° = 240°** and **480° = 120°**.
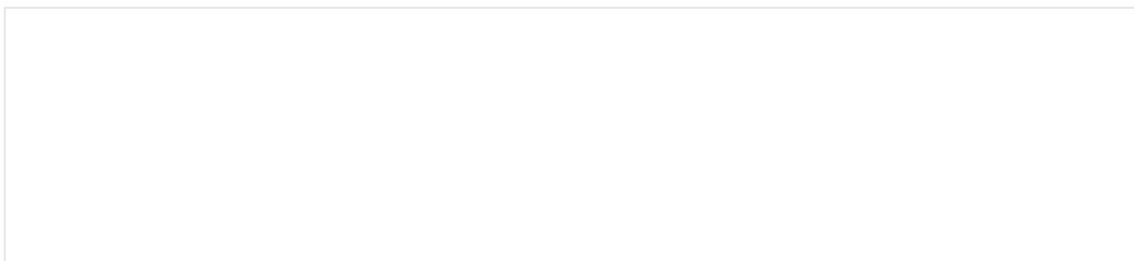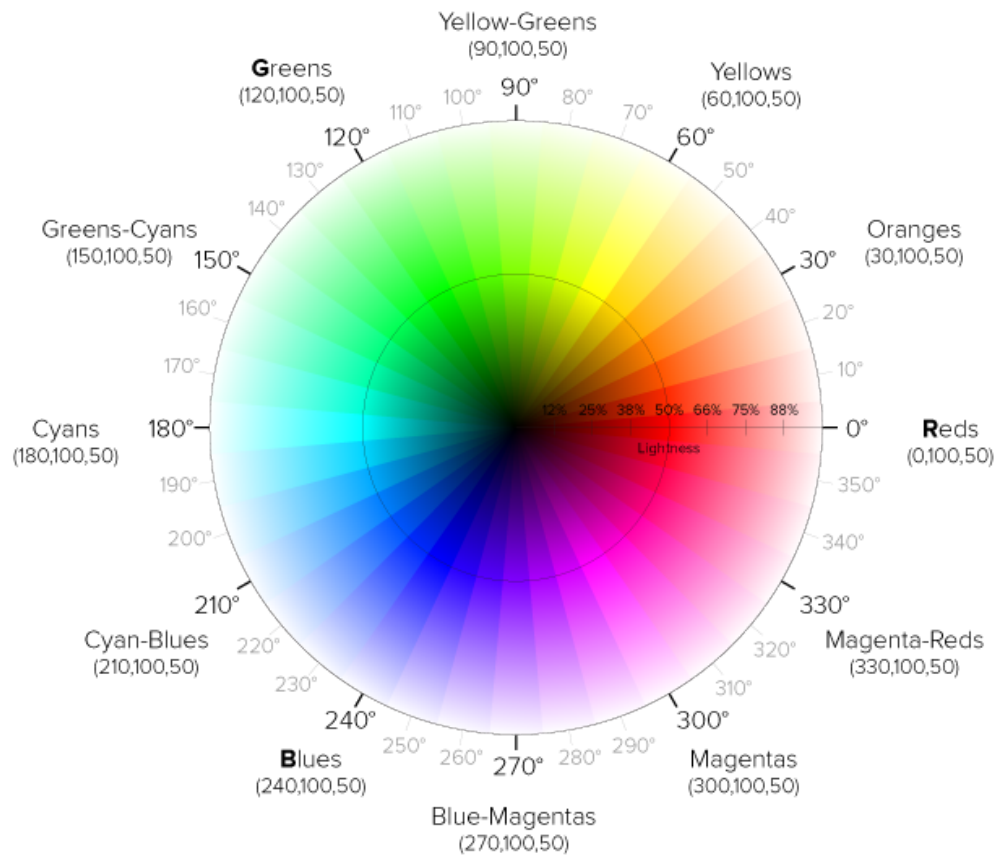
HSL Color Wheel by *Erin Sowards*.

Summarized,

- Saturation and lightness are represented as percentages

- 100% is full saturation, and 0% is a shade of gray

- 100% lightness is white, 0% lightness is black, and 50% lightness is "normal"

You set the color by picking an angle value and setting it as the value for the hue. You can then specify the saturation of your color. A saturation value of 0 (0%) gives a gray value (imagine de-saturating a colored photo, you'll end up with a black-and-white version of the photo with different shades of gray), whereas a saturation value of 1 (100%) means the color is fully saturated.

The third value, the lightness, lets you specify how "bright" the color should be. 0% means the brightness is 0, and the color is black. 100% means maximum brightness, and the color is white.

Lars Gunther has created a very nice interactive HSL color wheel. As the angle changes the color at the angle is shown at the center of the circle. You can click on any color angle to choose that color, and then tweak its lightness and saturation in the color picker below the color circle. It's a great tool to help understand how HSL colors work, so make sure to check it out.

If you pick a color and slowly increase its lightness from 0% to 100%, you will see the color changing from black, and then through a darker version of your color, to your color in its full brightness at 50%, and then getting brighter until finally, it reaches white at 100%.
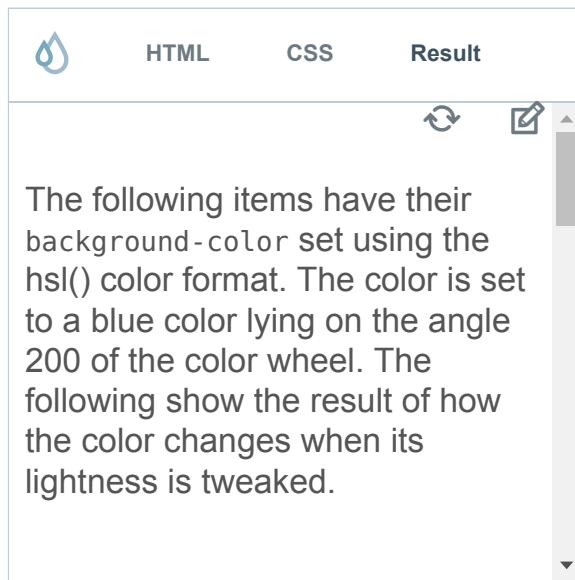
The advantage of HSL over RGB is that it is far more intuitive: you can guess at the colors you want, get the angle value, and then tweak to get different shades and tints. It is also easier to create sets of matching colors (by keeping the hue the same and varying the lightness/darkness, and saturation).

The following example sets the background color on the page to a light purple color. Check the color circle above to see which color is at a 270 degrees angle.

```
body {
    background-color: hsl(270, 100%, 50%);
}
```

The following demo shows the result of tweaking the lightness and saturation of the color blue (at angle 200 on the color wheel). Try

changing the values of the color, saturation and lightness to see how the background color of these boxes responds to the different values.

⬨	HTML	CSS	**Result**

🔄   ✎

The following items have their `background-color` set using the hsl() color format. The color is set to a blue color lying on the angle 200 of the color wheel. The following show the result of how the color changes when its lightness is tweaked.

Browser Support

The color values are supported in all major browsers: Chrome, Firefox, Safari, Opera, IE, and on Android and iOS.

The `hsl()`, `hsla()`, `rgba()`, `transparent`, and `currentColor` keywords are supported in Internet Explorer starting from version 9.

Further Reading

- [<color> on the Codrops CSS Reference](#)

- [CSS Color Module Level 3](#)

- [List of 147 supported colors in CSS](#)

- The CSS Transitions Module Level 3

Related Entries

- <color>

- hsla()

↑   E

Written by Sara Soueidan

Last updated March 6, 2016 at 3:36 am by Mary Lou

Do you have a suggestion, question or want to contribute? Submit an issue.