Working with colors on the web

This post contains my latest research on how to use color on the web, defining the different color spaces that modern browsers use and some that are part of the W3C color specifications but are not currently used in any browser.

I'll also take a look at some future standards that make the job of working with colors easier.

RGB Hex Colors

The first, and most used way to write colors in CSS, is the # character followed by a triplet of hexadecimal (base 16) values representing the red ®, green (g) and blue (b) channels for that particular color.

These two values are equivalent.

```
color: #00F;
color: #0000FF;
```

You can also use four-digit or an eight-digit notation to add an alpha value to indicate transparency.

If to use a blue color with 50% transparency we could do something like this:

```
color: #0000FF80;
```

The value for the alpha channel is different than what we'd use for RGBA or HSLA colors. It is not a value from 0 to 1 but a hexadecimal value from 00 to FF. So the middle point between 00 and FF is 80.

RGB and RGBA functions

The other way to use RGB colors is with the `rgb()` and `rgba()` functions. These work similarly to the hexadecima values but use 3 decimal values between 0 and

255 to control the RGB values.

To represent the blue color we use the following RGB function:

```
color: rgb(0, 0, 255);
```

To add transparency, we use the following RGBA function instead. RGBA adds a value between 0 and 1 or 0% to 100% that controls transparency.

```
color: rgba(0, 0, 255, 0.5);
color: rgba(0, 0, 255, 50%);
```

Named Colors

CSS offers a large set of `named colors` to make working with them easier to work with.

For a full list of the named colors see: named colors in the CSS Color Module Level 4 specification.

HSL HSLA

HSL colors are specified as a triplet of hue, saturation, and lightness.

The first argument specifies the hue angle.

In HSL the angle 0deg represents sRGB primary red (as does 360deg, and multiples of 360), and the rest of the hues are spread around the circle, as shown in HTML4 Color Keywords in HSL. Unless you know the color it may mean you'll have to play with the hue to get the exact color you want.

The next two arguments are the saturation and lightness.

For saturation, 100% is a fully-saturated, bright color, and 0% is a fully-unsaturated gray.

For lightness, 50% represents the "normal" color, while 100% is white and 0% is black.

If the saturation or lightness are less than 0% or greater than 100%, they are clamped to those values before being converted to an RGB color.

```
color: hsl(240, 100%, 50%);
```

If present in an HSLA color, the final argument specifies the alpha channel of the color. The value is the same as the fourth value in an RGBA color.

```
color: hsla(240, 100%, 50%, 1);
```

It takes a while to get used to the difference on how you write the colors and how to read them. But this is one more tool in the arsenal.

Lab and LCH colors

I became aware of Lab/LCH colors by reading Lea Verou's LCH colors in CSS: what, why, and how? and it has taken me a while to understand why this is important and why we should care.

When the web first came out computers were very different and the sRGB color space was more than enough for the 640x480 displays that we used to have.

But, as computers got faster and displays got crisper and got higher resolution, the sRGB colorspace is not enough to represent all the colors modern monitors can display.

Most modern monitors and some mobile displays work in P3 mode, which displays more colors that would normally be available to sRGB. As far as browsers are concerned, `display-p3` only works in Safari (desktop and iOS).

LCH stands for "Lightness Chroma Hue". The parameters are similar to, but not quite the same as, HSL colors.

The hue angles don't exactly match the hue in HSL colors.

In LCH, Chroma values are, theoretically, unbounded. LCH (like Lab) represents the full spectrum of human vision, and not all of these colors can be displayed on a

screen, even a P3 screen.

If you specify an LCH color that is not visible in a given monitor will be scaled down so that it becomes visible. This is similar to what happened with regular CSS colors when they were displayed in monitors with gamuts smaller than sRGB.

Browsers are beginning support for LCH/Lab colors.

- Safari is implementing it
- Chrome is starting to implement
- Firefox is discussing implementation

Until browsers fully implement lch colors we have to rely on tools like PostCSS and the postcss-lab-function. This tool ties you to the PostCSS workflow but if your code, like mine, uses Autoprefixer as a PostCSS module there is no major problem with introducing one more plugin to the process, particularly when it won't be user-facing (the plugin produces RGB or RGBA colors).

If you want to play with LCH/Lab colors, you can play with Lea Verou's lch color picker. This will give you an idea of the syntax and how LCH colors work.

Links and Resources

- CSS Color Module Level 4 — CSS WG
- CSS Color Module Level 5 — CSS WG
- 8-Digit Hex Codes? — CSS Tricks
- LCH colors in CSS: what, why, and how? — Lea Verou
- lch color picker — CSS Land
- Wide Gamut Color in CSS with Display-P3 — WebKit
- postcss-lab-function — Postcss
- <color> — MDN
- HTML4 Color Keywords in HSL — Eric Meyer
- HSL Color wheel demo — Lars Gunther
- An Easy Guide To HSL Color — Dudley Storey