Color in CSS

using new spaces, functions, and techniques to make your site shine

LULLABOT

Aubrey Sambor

Lead Engineer at Lullabot

Lives in Northampton, MA

Over 25 years of CSS experience!

Find me online

Mastodon
https://labyrinth.social/@starshaped

Blog
https://star-shaped.org

LinkedIn
https://www.linkedin.com/in/aubreysambor

Drupal.org
https://drupal.org/u/starshaped

LULLABOT

What will we talk about today?

- An overview of color spaces available today
- What's new in the CSS Color Module Levels 4 and 5
- New CSS color functions and how to use them
- Using CSS custom properties to change the values of color items within a color space (and a better way to do this!)
- Ways to make your site are more accessible by using light-dark and color-contrast

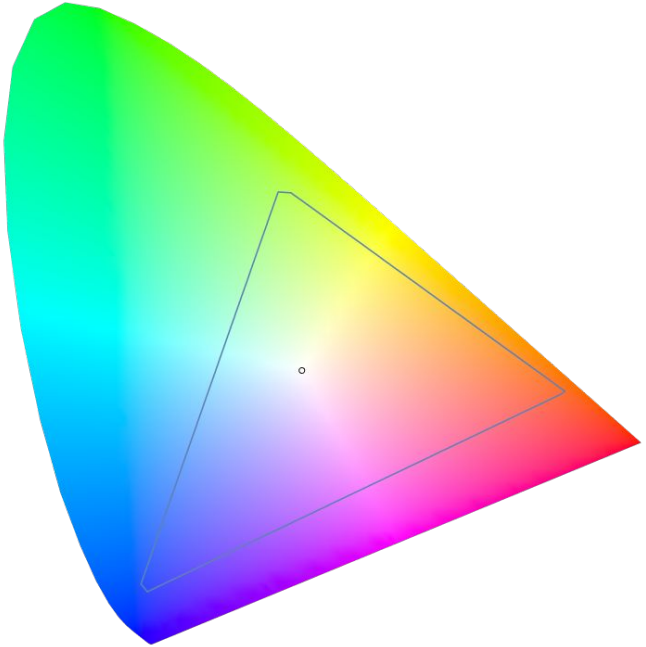
Current color spaces available today

First, what's a gamut?

Gamuts vs color spaces

- A **gamut** is a range of colors supported on a device, usually displayed as shapes within the human visual gamut (the range of colors that the human eye can detect).
- Examples of gamuts: sRGB, P3
- Most Apple devices support P3, also called wide gamut (as the P3 color range is much greater than the sRGB color range)

Gamuts vs color spaces



This is a view of the human visual gamut (HVG), the view of all colors visible by the human eye.

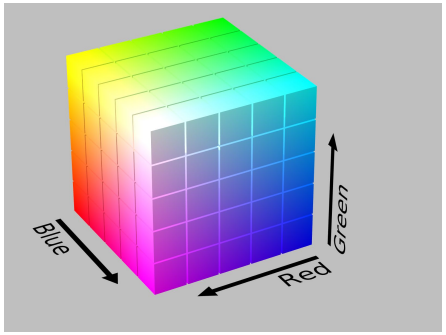
The triangle in the HVG represents the sRGB gamut, which is what the web has used up until CSS Color Module 4.

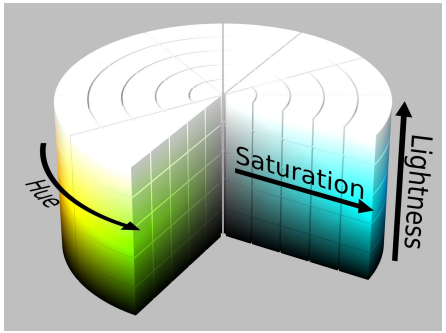
Other color spaces are also within the HVG, but have slightly different shapes depending on the colors they include.

Source: wikipedia & Chrome's CSS color guide

Gamuts vs color spaces

- A **color space** is the arrangement of colors within a particular gamut.
- Displayed in different shapes depending on the space
- For example, the RGB color space is often displayed as a cube, while the HSL color space is often displayed as a cylinder.





To summarize: a gamut is a collection of colors, while a color space is the arrangement of colors within a specific gamut.

Now, onto current color spaces!

The RGB color space

The RGB color space

- The color space you're probably the most familiar with
- The only color space you could really use up until CSS Color Module 4
- Consists of hex values, named colors, the rgb(), hsl(), and the new hwb() functions

Hex

The way most people first learned how to define color in CSS

```
.my-fun-color {
  color: #ff1493;
}
```

- Colors are defined using hexadecimal values from 0 to f, defined by `#RRGGBB`.
- Can also be written shorthand

```
.another-color {
  color: #000;
}
```

Named colors

A set of predefined keywords to represent hex colors

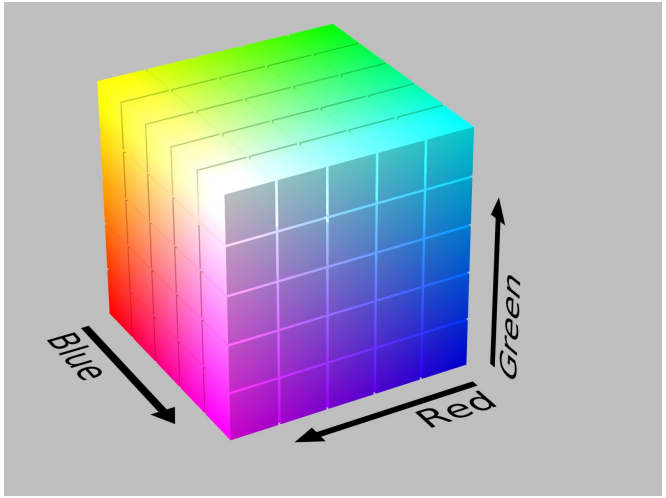
- Also includes currentcolor and transparent

```
.my-fun-color {
  color: blueviolet;
}
```

List of named colors on MDN

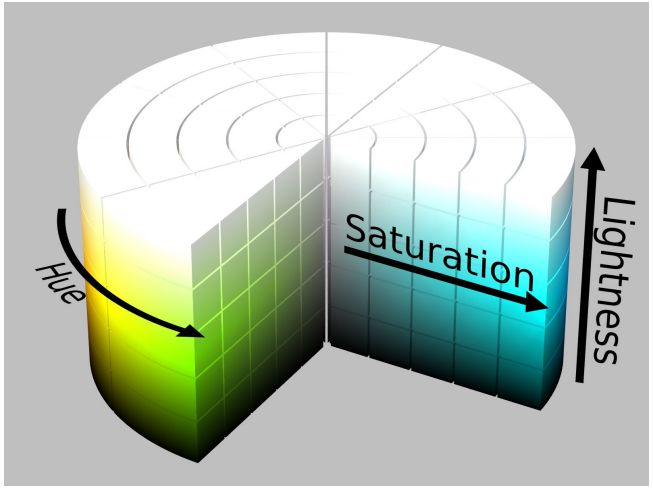
RGB

- Mixes red, green, and blue to make different colors
- Up until Color Module 4, uses the syntax `rgb(0,0,0)` with the 3 values accepting a range from 0 to 255 to represent each color in the space
- For opacity, use `rgba(0,0,0,50%)` with the 50% being the opacity
- More on these later!



HSL

- Arranges colors in the RGB space in terms of hue, saturation, and lightness in a color wheel
- Uses the syntax of `hsl(100deg,25%,25%)` with the hue as a degree on the wheel and saturation and lightness as percentages.
- For opacity, use `hsla(100deg,25%,25%,50%)` (can also depict the opacity as a decimal like .5)



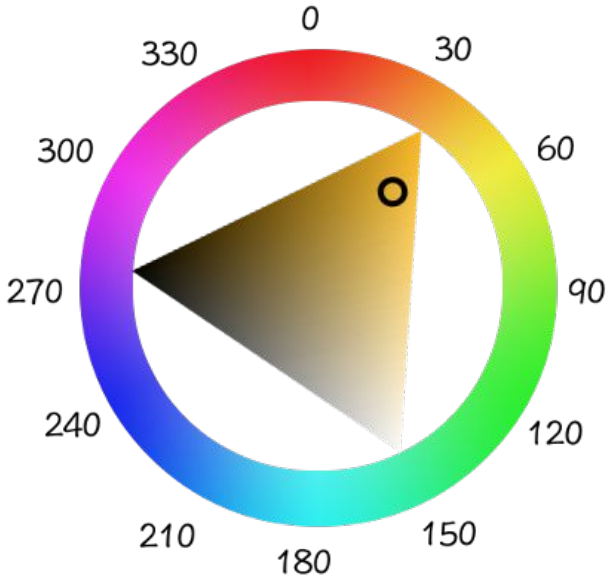
CSS Color Module Level 4

What's new?

- Support for color spaces other than RGB (CIE, Display P3, and others)
- New color functions: hwb(), lab() and oklab(), lch() and oklch(), color()
- Syntax updates for color definitions
  - `rgb(0,0,0)` can now be written as `rgb(0 0 0)` and opacity is no longer a separate function, written as `rgb(0 0 0 / 50%)`
- New named color: `rebeccapurple`

HWB

- New way to access the RGB color space
  - Hue, whiteness, blackness
- A lot like HSL, consists of an angle from the color wheel then percentages of white and black that mix into it.
- `hwb(100deg 25% 25%)`, `hwb(100deg 25% 25% / 50%)` for opacity
- (Note the new syntax with no commas!)



Graphic from https://www.quackit.com/css/color/values/css_hwb_function.cfm

The color() function

- After creating separate functions for rgb(), hsl(), hwb(), etc, a generic color() function was created that you can just pass parameters to
- Only uses red, green, and blue channels
- Uses decimals between 0 and 1, or percentages between 0 and 100% (and with the optional opacity definition)
- `color(display-p3 0 .5 .85)`

Codepen

The CIE color space

What is the CIE color space?

- Modeled after how colors are perceived by the human eye

- Perpetually uniform space
  - HSL's lightness isn't uniform across colors–in this Codepen example, the blue appears darker than the yellow even though the lightness is the same
  - In the CIE color space, the lightness will be uniform across all colors (which I'll show later!)

LAB

- Consists of lightness and two color channels (A and B don't stand for anything)--a range from red to green, and a range from blue to yellow.

- Written as `lab(58 -66 -30)`
  - Lightness represented as a percentage
  - Second parameter above 0 is more red, below 0 more green
  - Third parameter above 0 is more yellow and below 0 is more blue
  - Optional opacity `lab(58 -66 -30 / 50%)`

LCH

- Similar to LAB (it's also perpetually uniform), LCH stands for Lightness, Chroma, Hue
  - Lightness: Range from 0 to 100%
  - Chroma: Differs depending on the gamut, usually between 0 and .5
  - Hue: Same as HSL, range from 0 to 360 degrees

- Easier to visualize than LAB, as Chroma and Hue make more sense than A and B

- `lch(58% .32 241deg)`

OKLAB

- Pretty much the same as LAB, but with color corrections using math that makes my brain hurt
  - `oklab(58 -66 -30)`

- Gradients are more perpetually consistent than regular LAB due to said math and the color corrections make the colors display more accurately

  - In my example, the ‘blue’ in the LAB example displays more purple, while in OKLAB, it’s actually blue

Codepen

OKLCH

- Similar to LCH in the same way that OKLAB is similar to LAB with similar color corrections to fix the blue hue shifts
- Has a color picker! https://oklch.com
- LCH and OKLCH colors will display differently due to different hue angles
- `oklch(58% 32 241deg)`

Codepen

CSS Color Module Level 5

What's new?

- The color-mix() function
  - The ability to mix colors within a given color space
- Relative color syntax
  - An easier way to manipulate colors using less code
- The light-dark() function
  - Simpler way to change colors based on light or dark mode
  - More about this one later...

color-mix()

- Gives the ability to mix colors using CSS in an easy way without Sass or custom properties with HSL
  - `--mix-srgb: color-mix(in srgb, blue, white);`
- Can mix colors in different color spaces!
  - `--mix-oklch: color-mix(in oklch, blue, white);`
- Weird color issues using color-mix in the OKLCH color space, however...
  - Github issue discussing how to fix it

Codepen

Before we talk about relative color syntax...

Updating colors using custom properties

LULLABOT

Using custom properties with color

- To create a variant of a color, you've had to create a custom property for each part, manipulate the one part, then combine them all together again.
- Works best using HSL but you can probably do this in HWB too!
- Doing it this way is great, but… what if I told you there was a BETTER way?

Codepen

NOW it’s relative color syntax time!

LULLABOT

Relative color syntax

- A more concise way to manipulate color!
- Supported in Chrome and Safari currently (come on, Firefox!)
- ```
  .relative-syntax {
    background-color: hsl(from green h s l);
  }

  relative-syntax-hue-shift {
    background-color: hsl(from green calc(h + 200) s l);
  }
  ```

Codepen
  ```

Color functions to help with accessibility

light-dark()

- Simpler way to add styles according to whether you're using dark mode or light mode on your system
- Currently supported only in Firefox
- Declared in the :root of your CSS file
  - `color-scheme: light dark;`
  - `--text-color: light-dark(#333, #ccc);` (first is light mode, second is dark)

Codepen

color-contrast()

- Experimental feature, only supported behind a flag in Safari
- Part of Color Module Level 6
- First color is the background color, then two text colors
- Text color will change depending on which has the ‘most’ contrast with the background color! (Uses the WCAG 2.1 algorithm but this may change)
- `color: color-contrast(#cf8a68 vs #ffffff, #000000);`

Codepen (by Dave Rupert)

Wrapping up...

Wrapping up

- There are so many new things happening with color in CSS today!
  - New color spaces such as HWB, LAB, and OKLCH
  - New functions such as color(), color-mix(), and so much more
  - Easier ways to manipulate color variations using relative color syntax (Firefox just needs to support it first!)
  - Functions to help with accessibility such as light-dark() and color-contrast()

Further reading

- https://developer.chrome.com/docs/css-ui/high-definition-css-color-guide
- https://12daysofweb.dev/2022/css-color-spaces-relative-color-syntax/
- https://developer.mozilla.org/en-US/blog/css-color-module-level-4/
- https://www.smashingmagazine.com/2021/11/guide-modern-css-colors/
- https://www.bram.us/2023/10/09/the-future-of-css-easy-light-dark-mode-color-switching-with-light-dark/
- https://css-tricks.com/the-expanding-gamut-of-color-on-the-web/
- https://evilmartians.com/chronicles/oklch-in-css-why-quit-rgb-hsl
- https://www.w3.org/TR/css-color-4/
- https://www.w3.org/TR/css-color-5/

Aubrey Sambor

Lead Engineer at Lullabot

Lives in Northampton, MA

Over 25 years of CSS experience!

Find me online

Mastodon
https://labyrinth.social/@starshaped

Blog
https://star-shaped.org

LinkedIn
https://www.linkedin.com/in/aubreysambor

Drupal.org
https://drupal.org/u/starshaped

LULLABOT

Tada! 

LULLABOT