Colour (Color)

CSS3

Colour and the Web

It is not uncommon that you'll need to adjust the colour value of text, an element or some part of an element as you build and style websites. There are a few options to pick from when writing such CSS rules.

So far, we've just used colour keywords. There is an extensive list of colour keywords available on the Mozilla Developer Network website, if you're curious what you can choose from.

Despite the dozens options available via colour keywords, using only colour keywords is extremely restrictive. If you need to match your website to brand colours, or just want it to look a little more unique, you'll need types of values that are much more flexible.

Keep in mind the modern digital world (computer monitors, laptop screens, phone screens, etc.) is meant to display upwards of sixteen million different colours, not dozens. This is important especially where photographs and graphics are concerned.

RGB

The web follows the Red Green Blue (RGB) colour model to describe colours based on a combination of three base channels (as the name suggests.) See an interactive visual representation on W3Schools.

Most verbosely this can be communicated in CSS in functional notation via the RGB syntax, wherein instead of “red” you can type rgb(255, 0, 0). The three values inside of the parentheses may be any integer value ranging from 0 to 255 (giving you a total of 256 possible intensities for each of red, green, and blue respectively.) Each of the three numbers represents, in order: red, green, and blue.

```
/* Red Text */
color: rgb( 255, 0, 0 );
/* Green Text */
color: rgb( 0, 255, 0 );
/* Blue Text */
color: rgb( 0, 0, 255 );
/* White Text */
color: rgb( 255, 255, 255 );
/* Black Text */
color: rgb( 0, 0, 0 );
```

Hexadecimal Notation

Faster to type, and just as—if not more frequently—used across the web is the hexadecimal notation. It also represents 256 increments of intensity for each of red, green, and blue.

The format is an octothorpe (#) followed by three two-digit values: #RRGGBB

Each two-digit value is composed of values ranging from 00 (the lowest) to ff (the highest.)

```
/* Red Text */
color: #ff0000;
/* Green Text */
color: #00ff00;
/* Blue Text */
color: #0000ff;
```

```
/* White Text */
color: #ffffff;
/* Black Text */
color: #000000;
```

If the colour exhibits repeated characters, it can be shortened from six to three digits like so:

```
/* Red Text */
color: #f00;
/* Green Text */
color: #0f0;
/* Blue Text */
color: #00f;
/* White Text */
color: #fff;
/* Black Text */
color: #000;
```

See an interactive visual representation on W3Schools.

Hue Saturation Lightness (HSL)

HSL colours describe colour a little differently, and is sometimes favoured by designers. It defines colours based on hue, saturation, and lightness.

There is an interactive visual representation of HSL found on W3Schools that may help with this.

- Hue
  The angle, in degrees, of the colour within the colour wheel. Values ranging from 0 to 359 are accepted.
- Saturation
  A percentage representing the saturation of the selected hue.
- Lightness
  A percentage representing how bright or dark the hue will be. 100% is white, 0% is black.

```
/* Red Text */
color: hsl( 0, 100%, 50% );
/* Green Text */
color: hsl( 120, 100%, 50% );
/* Blue Text */
color: hsl( 240, 100%, 50% );
/* White Text */
color: hsl( 0, 0%, 100% );
/* Black Text */
color: hsl( 0, 0%, 0% );
```

Converting Between Formats

The easiest way to find the representation of a colour in each format is to use a colour conversion tool.

W3Schools offers one such tool, but be aware that there are many other options out there (and some browsers include these out of the box or via an extension.)

Alpha Channel

There is one more part to colour that is important—the alpha channel!

Alpha values represent how transparent, translucent, or opaque something should be. Sometimes you'll need a background colour, border colour, etc. to be a bit (or entirely) see-through.

Alpha typically appears as an additional value at the end of these previously covered syntaxes.

For example:

```
/* Opaque Red */
rgb( 255, 0, 0 );
/* 50% Transparent Red */
rgba( 255, 0, 0, 0.5 );
```

```
/* Opaque Red */
hsl( 0, 100%, 50% );
/* 50% Transparent Red */
hsla( 0, 100%, 50%, 0.5 );
```

```
/* Opaque Red */
#ff0000;
/* 50% Transparent Red */
#ff000050;
```

Note that most opacity values in CSS range from 0 (transparent) to 1 (opaque.) Therefore, 50% is represented by 0.5, and so on.

Reusable Colours

As you style more and more pages, you'll often come across cases where you end up re-typing (or copy-and-pasting) a colour dozens—if not hundreds—of times.

In modern CSS there is a solution to this: custom properties / variables.

Essentially this feature offers a way to assign a value to a name. Each time you call that name, the value will be used in its place. In this way you can change the value in one place (typically at or near the top of your file) instead of all of them as you work on your styles. Experimentation and replacement becomes much more efficient via this method. Custom property names are always prepended and marked via use of two hyphens (--).

Try adding the following at the top of your file to define a value for the "--text-color" name:

```
:root {
    --text-color: rgb( 255, 0, 0 );
}
```

Once the name is set, you can use it elsewhere via the var function:

```
h1 {
    color: var( --text-color );
    Border-bottom: 2px solid var( --text-color );
}
p { color: var( --text-color ); }
```

Other Colour-Related Topics

You may also want to explore what CSS can do via the following…

- Gradients
- Opacity
- Transparent Keyword
- CurrentColor Keyword

Recommended Reading

Continue exploring colour on the web with the following reading:

- Meyer, E. A; Weyl, E. (October 2017). CSS: The Definitive Guide, 4th Edition. O'Reilly Media, Inc.
  - Chapter 4. Values and Units
    - Color