Advanced Colors

We already know that colors can be defined by name, RGB, or hex values, but CSS 3 also allows you to paint away with **HSL** - hue, saturation, and lightness - as well as stipulating **transparency**.

There are no super special properties at play here - **HSL** and **RGBa** (the "a" standing for "alpha", as in "alpha transparency") can be applied to any property that has a color value, such as `color`, `background-color`, `border-color` or `box-shadow`, to name a mere handful.

Alpha transparency

RGBa opens up an exciting new dimension to web design, allowing you to set the transparency of a box or text. If you wanted a smidgen of a snazzy background image to peep through a heading, for example, you might use something like this:

```
h1 {
    padding: 50px;
    background-image: url(snazzy.jpg);
    color: rgba(0,0,0,0.8);
}
```

A standard value of `rgb(0,0,0)` would set the heading to pure black but that fourth value, in `rgba`, sets the level of transparency, "1" being completely opaque, "0" being completely transparent. So `rgba(0,0,0,0.8)` is saying red="0", green="0", blue="0", alpha="0.8", which, all together, makes it 80% black.

This doesn't only apply to text, of course, you could apply a transparent background color to an entire box, a transparent box shadow... anywhere where you can use `rgb`, you can used `rgba`.

Hue, saturation, and lightness

Color names aside, web colors have always been red-green-blue biased, be that through hex codes or explicit RBG (or RGBa). Although mildly less straightforward (especially if your brain is trained to break down colors into red, green and blue), HSL can actually be more intuitive because it gives you direct control over the aspects of a color's shade rather than its logical ingredients.

It is used in a similar way to rgb:

```
#smut { color: hsl(36, 100%, 50%) }
```

Rather than each sub-value being a part of the color spectrum, however, they are:

- **Hue** ("36" in the above example): Any angle, from 0 to 360, taken from a typical color wheel, where "0" (and "360") is red, "120" is green and "240" is blue.
- **Saturation** ("100%" in the example): How saturated you want the color to be, from 0% (none, so a level of grey depending on the lightness) to 100% (the whole whack, please).
- **Lightness** ("50%" in the example): From 0% (black) to 100% (white), 50% being "normal".

So the example used here will produce an orange (36°) that is rich (100% saturation) and vibrant (50% lightness). It is the equivalent of `#ff9900`, `#f90`, and `rgb(255, 153, 0)`.

HSLa

Hey, man, this funky fresh transparency and HSL can be combined?! You'd better believe it. Here's **HSLa**:

```
#rabbit { background: hsla(0, 75%, 75%, 0.5) }
```

You can figure out what that does, right?

`hsl` and `hsla` are supported by most modern browsers, including IE versions 9 and above.

Related pages

- Next Page: At-Rules
- Previous Page: Universal, Child, and Adjacent Selectors
- Colors (CSS Beginner Tutorial)

Working Examples

- Box shadows: with RGBa colors
- CSS transitions: using `border-radius` and RGBa colors