

During my first week at Astonish Design, I was fortunate enough to attend an [Austin Drupal Users Group meetup](#) where [Ian Carrico](#) spoke about [Power Themeing](#). Power Themeing, as Ian coined it, can be used by anyone who does theme work and writes CSS for any platform. This new way of thinking has catapulted me from the days of relying on someone else's responsive framework (Twitter Bootstrap, HTML5 Boilerplate, etc), to writing my own in the same amount of time. Frameworks can be powerful, but I fear that they are holding back innovation.

In the current age of [Responsive Web Design \(RWD\)](#) and multiple browsers, the issue has been writing lengthy css, with multiple media queries, that respond to browser size and work everywhere or fall back cleanly (non-CSS3 browsers). Let's not even mention the browser specific css:

```
.gradient {
    *zoom: 1;
    -ms-filter: "progid:DXImageTransform.Microsoft.gradient(gradientType=0, startColorstr='#FFAAAAAA', endColorstr='#FFEEEEEE')";
    filter: progid:DXImageTransform.Microsoft.gradient(gradientType=0, startColorstr='#FFAAAAAA', endColorstr='#FFEEEEEE');
    background: #cccccc;
    background: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, #aaaaaa), color-stop(100%, #eeeeee));
    background: -moz-linear-gradient(top, #aaaaaa, #eeeeee);
    background: -o-linear-gradient(top, #aaaaaa, #eeeeee);
    background: linear-gradient(top, #aaaaaa, #eeeeee);
}
```

Compass

CSS as a language.

Enter [Compass](#) with its wonderful extensions. Compass is a, Ruby based, CSS authoring framework that uses Sass. [Sass](#) allows for variables, mixins, nesting, inheritance, and much more. These features reduce the amount of CSS you need to write. [LESS](#) is similar to Sass, but falls short because of it's lack of Compass integration. Compass takes Sass a step further. Compass adds multitudes of useful and intuitive functions. In addition, Compass can be set to watch for changes and automatically compile you Sass to CSS. See <http://compass-style.org/help/> for a good intro video. With Compass and Sass our above background gradient CSS now looks like:

```
@import "compass"; //called once

.gradient {
    /* IE; docs say this should go first (or better, placed in separate IE-only stylesheet): */
    @include filter-gradient(#aaaaaa, #eeeeee);
    /* Fallback: */
    background: #cccccc;
    /* CSS 3 plus vendor prefixes: */
    @include background(linear-gradient(top, #aaaaaa, #eeeeee));
}
```

*** I would like to point out that the '@include filter-gradient' uses the same colors, not IE specific ones. Also, comments are

ignored during processing.

Another extremely useful aspect of Compass is [Sass partials](#). Partials, named `_file.scss`, are easily called in include statements and are compiled into one CSS file. You can have dozens of Sass files to keep your Sass broken up how you like, but one CSS. Developers can work on different files at the same time. Features or sections of a site can be broken up in separate files. Your workflow can dictate your own Sass partial structure.

Compass Extensions

A few Compass extensions that I know about:

[Sassy Buttons](#) - Sassy Buttons is a Compass extension that adds functions for super simple pure CSS buttons:

Sass:

```
.btn-test{
  @include Sassy-button-gradient("matte", #ff6f13);
}
```

CSS:

```
.btn-test {
  *zoom: 1;
  filter: progid:DXImageTransform.Microsoft.gradient(gradientType=0, startColorstr='#FFFF6F13', endColorstr='#FFAC4300');
  background: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, #ff9e5f), color-stop(70%, #ff6f13));
  background: -webkit-linear-gradient(#ff9e5f, #ff6f13 70%);
  background: -moz-linear-gradient(#ff9e5f, #ff6f13 70%);
  background: -o-linear-gradient(#ff9e5f, #ff6f13 70%);
  background: linear-gradient(#ff9e5f, #ff6f13 70%);
  -webkit-box-shadow: #ffccac 0 1px 0 inset;
  -moz-box-shadow: #ffccac 0 1px 0 inset;
  box-shadow: #ffccac 0 1px 0 inset;
  border: 1px solid #f96100;
  color: white;
  text-shadow: #d75400 0 -1px 0;
}
.btn-test:hover {
  text-decoration: none;
}
.btn-test:hover {
  background-color: #f96100;
  background: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(3%, #ff8e46), color-stop(75%, #f96100));
  background: -webkit-linear-gradient(#ff8e46 3%, #f96100 75%);
  background: -moz-linear-gradient(#ff8e46 3%, #f96100 75%);
  background: -o-linear-gradient(#ff8e46 3%, #f96100 75%);
  background: linear-gradient(#ff8e46 3%, #f96100 75%);
}
.btn-test:active {
  -webkit-box-shadow: #e95b00 0 1px 2px inset;
  -moz-box-shadow: #e95b00 0 1px 2px inset;
  box-shadow: #e95b00 0 1px 2px inset;
}
```

```

}
.btn-test.disabled, .btn-test[disabled] {
  opacity: 0.6;
  background: #ff6f13;
  cursor: default;
  -webkit-box-shadow: none;
  -moz-box-shadow: none;
  box-shadow: none;
}
/**** BUTTON HTML HERE ****/

```

[Breakpoint](#) / [Respond-To](#): Breakpoint allows you to easily define and call breakpoints in your CSS, but Respond-To allows you to semantically define and call numerous breakpoints. See: <https://github.com/snugug/respond-to#usage-example> Comparison: <https://github.com/snugug/respond-to#comparison-with-breakpoint-extension>

Needless to say these extensions make responsive CSS super simple. Just call the function and the CSS will only be used at that breakpoint. No longer does your CSS need to be separated by media queries.

```

$breakpoints: 'full navigation, large' (700px 800px),
              'monochrome screen' (monochrome),
              'monochrome portrait, wide' (700px, orientation portrait, monochrome);

.respond {
  //normal CSS here
  @include respond-to('full navigation', 'large') {
    //Conditional CSS Here
  }
}

```

[Susy](#): Susy is a Compass/Sass responsive grid extension with a complete responsive framework. See: <http://susy.oddbird.net/guides/getting-started/>
 How to blog post: <http://sectionr.be/blog/>
 Cool things coming in the future for Susy: <http://oddbird.net/2013/01/01/susy-next/>

Here is a complete list of Compass extensions (unvetted):
<https://rubygems.org/search?query=compass>

Drupal + Sass & Compass

Lets get Drupal specific to close.

All of the Compass goodness that has been mentioned so far has been rolled into a Drupal base theme called [Aurora](#), which is maintained by Ian and [Sam Richard](#). Aurora also has a distribution called Corona that is built on Susy. Corona runs as a sub theme of Aurora and can be deployed with one command via terminal. It comes fully laid out with theme structure and Sass partials in place. See [Ian's blog post](#) for more about how to use Corona. Aurora has some exciting features like Modernizr and view port overlay display for themeing help, it's own Compass extension, and

some promising JavaScript features in the works. In addition, you can enable the [Borealis Suite](#) for responsive images and semantic blocks.

References in short:

Power Theming: <http://chinggizkhan.github.com/power-theming-drupal/>

Sass: <http://sass-lang.com/>

Compass: <http://compass-style.org/>

Sassy Buttons: <http://jaredhardy.com/sassy-buttons/>

Respond-To: <https://github.com/snugug/respond-to>

Aurora: <http://drupal.org/project/aurora>

Compass, Aurora, and Corona... Oh My!:

<http://fourkitchens.com/blog/2012/11/13/compass-aurora-and-corona-oh-my>

Borealis Suite: <http://drupal.org/project/borealis>