Second Edition

Aurora

a sass+compass based html5 base theme for drupal

Sam Richard
Front End Developer

Welcome

Thank you for using Aurora, an HTML5, Sass and Compass powered, responsive optimized, mobile first base theme designed for people who like theming, not pointing and clicking. Base themes are powered by the Aurora Compass extension, providing a collection of awesome tools for you to pick up and run with to create custom grids, progressively enhanced future friendly websites, and reusable and maintainable media queries. Plus, you can tap into all of the other amazing Compass extensions available!

Aurora has a companion module, the Borealis Suite, that includes a module for semantic blocks and a responsive image solution That Just Works™. Aurora also has support for the Conditional Stylesheets module, and I highly recommend Fences for custom and semantic field templates and Elements for HTML5 form elements, so be sure to grab those as well!

Aurora and Borealis are the brainchild of Sam Richard. He is Snugug throughout the Internet, tweets from @Snugug, and can be pinged at Snugug in many of the Drupal IRC rooms. Feel free to reach out if you have any questions.

Installing Sass, Compass, and Aurora

Sass, Compass, and the Aurora Compass extension are the real power behind Aurora sub themes, so we need to install them.

We are going to be using the Command Line, but don't be scared, it'll be easy!

Let's get you started!

Mac Installation

Easy Peasy Lemon Squeezy!

Mac users, you have it off easy! All OSX installs come with Ruby Gems installed by default, so it's super easy to install Sass, Compass, and Aurora. Open up your Terminal (either search for it using Spotlight or an application launcher like Alfred, or by opening it manually, by default located in Macintosh HD->Applications->Utilities) and typing in the following commands, pressing return after each command. You will be asked to type in your password the first time you do this:

```
sudo gem install compass
sudo gem install sass
sudo gem install compass-aurora
```

That’s it! You’re done!

Linux/Unix Installation

Building from Source

Unix and Linux do not usually come with Ruby/Ruby Gems installed, and as such you're going to need to install them yourselves. If you are not on a system that supports apt-get, follow the instructions from Ruby Gems:

Apt-Get ALL The Things!

If you are on a Unix/Linux system that supports apt-get, the instillation of Ruby and Ruby Gems is fairly easy. Open up your command line and apt-get the following and write the gem's path to PATH:

```
sudo apt-get install ruby-full build-essential
sudo apt-get install rubygems
export PATH=/var/lib/gems/1.8/bin:$PATH
```

Installing Sass, Compass, and Aurora

Once you have Ruby Gems installed, or if you had Ruby Gems installed already, installing Sass, Compass, and Aurora is very easy! Pop back into your command line, and type the following:

```
sudo gem install compass
sudo gem install sass
sudo gem install compass-aurora
```

Windows Installation

Get a Drink, We're Gonna Be Here a While

Getting Sass, Compass, and Aurora set up on a Windows computer is a little bit harder than on a Mac or Linux/Unix because Ruby Gems is a little harder to get running. In addition to installing Sass, Compass, Aurora, and Ruby Gems, we are also going to install Cygwin to make working in the command line easier and make the commands throughout this book work for you as well.

Installing Cygwin

Cygwin gives your Windows system a Unix-like command line interface, which will make your life developing on Windows easier and the commands used throughout this manual work on your computer. Cygwin is free and open source project. See the official documentation on how to install and use Cygwin.

Installing Ruby Gems

Installing Ruby and Ruby Gems on a Windows computer is surprisingly not too hard, simply go to Ruby Installer, download the current version, and follow the setup wizard.

Installing Sass, Compass, and Aurora

Once you have Cygwin and Ruby installed, installing Sass, Compass, and Aurora is very easy! Open up Cygwin and type the following:

```
gem install compass
gem install sass
gem install compass-aurora
```

Creating Your Subtheme

2

This is great, how do I get started you say? Well look no further!

A New Theme Is Born!

A Compass Powered Starterkit

The starterkit for Aurora subthemes is a little bit different than other Drupal base themes; there is no starterkit in the theme download, it's entirely driven by Compass. We're going back into the command line, and we're going to be using some basic Unix commands. Thanks to Cygwin, these commands should work on all three major platforms.

Choose Your Grid System

Before creating your subtheme, you're going to need to choose which grid system you want to use. Aurora comes in three flavors with two grid system, the Susy flavor, the Singularity flavor, and the gridless flavor.

Build Your Subtheme

After you've chosen your grid system, the first thing we're going to do is open up our command line tool and change directory to our theme directory.

```
cd ~/Sites/drupal/sites/all/themes
```

Once there, it's a simple command to create your subtheme. In each of these commands, replace <my_theme> with the machine name of your theme. This means that the name can only contain letters, numbers, and underscores. So! Let's create your subtheme! I'm going to abbreviate *compass create* as cc, but you're going to need to write the whole thing.

For no grid system, or to use your own:

```
cc <my_theme> -r aurora --using aurora
```

For the Susy grid system:

```
cc <my_theme> -r aurora --using aurora/susy
```

For the Singularity grid system:

```
cc <my_theme> -r aurora --using aurora/singularity
```

All three of these will create you your new subtheme with all of the awesomeness that you need to get started!

Getting To Know Aurora

3

What makes Aurora, Aurora? How does it do the crazy things it does? Learn all this and more in here!

Anatomy of your Subtheme

Files In The Aurora Starterkit

What's All This Then?

An Aurora subtheme comes with lots of goodies, and sorting through all of them can be a bit overwhelming. Let's go over them to make it less so.

Subtheme Info File

This is your subtheme's .info file; what tells Drupal that your theme exists. Besides the basics, the .info file also includes support for the Conditional Stylesheets module for the Internet Explorer stylesheet, the available regions in your new theme, and a handful of options for Aurora. See the next section, Aurora Options, for details on those options.

template.php

This is a starter template file that includes an HTML Preprocess hook that adds in your subtheme's Modernizr build as well as loader.js to your site.

modernizr.js

To quote Modernizr's Website:

> Modernizr is an open-source JavaScript library that helps you build the next generation of HTML5 and CSS3-powered websites.

At its very basic, Modernizr checks for browser capabilities and adds classes to the HTML tag of the page per HTML5/CSS3 capability of the current browser as well as adds a JavaScript object describing the browser capabilities, thus allowing for easy progressive enhancement by targeting the classes with your CSS and the JavaScript object in your JavaScript. Aurora is using a custom Modernizr development build that incudes HTML5 Shiv and Modernizr.load, also known as yepnope. BE SURE TO CREATE A PRODUCTION BUILD BEFORE GOING LIVE!

loader.js

loader.js is a file designed to hold your yepnope file loading options. By default, it comes with a test for Touch and loads in Hammer.js

hammer.js

Hammer.js is a JavaScript library for multi-touch gestures. It's a small standalone library designed specifically as an easy way of implementing touch events.

config.rb

config.rb is your Compass configuration file. It holds configuration information for Compass and generally only needs to be edited if you want to add a Compass extension to your theme or change the output style.

If you would like to change the output of CSS, it's as simple as changing the output_style in config.rb. By default, the output style is :expanded, but it can be set to any of the following: :expanded, :nested, :compact, or :compressed. If you would like your output CSS to have comments above each selector telling you what line of which partial it comes from, set line_comments to true config.rb.

style.scss, print.scss, ie.scss

style.scss, print.scss, and ie.scss are Sassy CSS formatted Sass files located in the sass folder of your subtheme. Aurora is built using an object-oriented approach to Sass partials, and as such, these files should not hold any styles itself, but rather links to partials that, when the file gets compiled, will pull in selectors from the various partials. The @import statements you see in this file are not CSS imports but rather compile time imports, meaning they do not have any effect on CSS load time. These files will each print out to style.css, print.css, and ie.css.

Global Partials

The files in the Global partials folder are designed as items to be shared between style, print, and ie sass files. These include all of your Compass imports, variables, functions, mixins, and extendables.

Design, Layout, and Style Guide

In the Partials folder, you'll see Design, Layout, and Style Guide folders each with containing style, print, and ie partials. These partials are well documented and are designed to allow you to build a Style Guide driven website as well as provides for a way of separating layout from design to ease maintainability.

Aurora Options

Enhancements Overview

Chrome Frame

Aurora can add Google's Chrome Frame for any non-supported version of Internet Explorer. This can be configured by your theme's settings page or by setting *settings[aurora_enable_chrome_frame] = 1* in your theme's .info file to enable Chrome Frame and *settings[aurora_min_ie_support] = 10* to choose the minimum supported version of Internet Explorer you support.

Footer JavaScript

Aurora, by default, moves all JavaScript that is loaded into your page using drupal_add_js to the bottom of the page in an effort to decrease page load time and prevent JavaScript from blocking the rendering of your pages. Because functionality of Modernizr is needed as a bare minimum for our pages to work correctly, it is the only piece of JavaScript that, by default, gets loaded in the header of our page. If you would like your JavaScript to be printed in the header, include the *'force header' => true* option in your drupal_add_js or hook_js_alter'd javascript. This can be configured by your theme's settings page or by setting *settings[aurora_footer_js] = 1* in your theme's .info file.

HTML Tag Pruning

Nathan Smith has proposed pruning HTML tags to remove the XHTML cruft. This can be configured by your theme's settings

page or by setting *settings[aurora_html_tags] = 1* in your theme's .info file.

Theme Registry Rebuild

During the development process, it may be useful to rebuild the theme registry on page reload. This is a significant performance hit and should only be enabled while developing. This can be configured by your theme's settings page or by setting *settings[aurora_rebuild_registry] = 1* in your theme's .info file

LiveReload Integration

LiveReload is an awesome tool that will do a soft reload of your page's assets on change, allowing for super easy design in browser. This will add LiveReload's activation script straight into your theme, thus allowing you to use LiveReload even in places where they don't have a browser extension. This can be configured by your theme's settings page or by setting *settings[aurora_livereload] = 1* in your theme's .info file

The Awesomeness of Sass

4

Aurora is built on the awesomeness of Sass using the Compass framework. Come on, get Sassy.

Compass Extensions

Included Compass Extensions

Compass Extensions

Compass extensions are similar to Drupal modules, allowing users to tap into the Sass front end community and use what others have made to make your life easier! Below are the Compass extensions that come installed with the Aurora Compass extension that powers your subtheme. They all do awesome things in a way that allows you to have full control, making them even better than any CSS framework you may otherwise use.

Susy

Susy is a tried and true, very stable, awesome symmetric grid system for Sass. It is a semantic fluid grid system that is designed to allow you to create your own custom grids. It is one of two grid systems offered as default options in Aurora.

Singularity

Singularity is a new grid system focused on creating asymmetric and odd grids for Sass. It is a semantic fluid grid system that is designed to allow you to create your own custom grids. It is one of two grid systems offered as default options in Aurora..

Toolkit

Toolkit is a set of mixins, functions, and extendables for Responsive Web Design and Progressive Enhancement.

Sassy Math

Sassy Math is a set of advanced math functions for Sass.

Modular Scale

Modular Scale is a Compass extension that implements Tim Brown's More Meaningful Typography math in easy to use Sass functions.

Breakpoint

Breakpoint is a system for Sass that allows for DRY, reusable media queries including options for no query fallbacks and development options to get the context of a media query from within functions and mixins.

Respond-to

Respond-to provides more semantic naming of media queries for Breakpoint.

Sassy Buttons

Sassy Buttons provides a simple way to create fancy CSS3 buttons.

Compiling Your CSS

All Dressed Up With Nowhere To Go

All Dressed Up With Nowhere To Go

Once you have all of your variables set and are ready to start theming your site, you're going to need to get Compass compiling your Sass into CSS for you! There are two approaches, either through an application, or straight through the terminal. The former approach is better for those who are afraid of the terminal but offers less power. The terminal approach can be a little bit daunting at first, but is the more powerful way to start compiling your CSS.

The Terminal Approach

This is where we jump back into the command line. Open up your Terminal on your Unix/Linux or Mac computer or Cygwin on your Windows computer and navigate to your subtheme's directory. This list of Basic Terminal Commands will help you navigate to your subtheme.

```
cd ~/Sites/drupal/sites/all/themes/my_theme
```

Make sure you are in the same directory as your config.rb file by listing the contents of your current directory:

```
  ls
  > config.rb javascripts sass stylesheets
template.php my_theme.info
```

Great, I see config.rb. To start Compass watching your subtheme for changes, simply enter *compass watch*

```
  compass watch
  >>> Compass is polling for changes. Press Ctrl-C to
Stop.
```

That's it! Whenever you save a Sass file in your theme, Compass will see that, try and compile it and, if successful, overwrite your CSS file and, if not, tell you the error and have you go fix it. To stop Compass from watching for changes, either press Ctrl+C on your keyboard or close your terminal window.

The Application Approach

There are a few applications that will compile Sass into CSS for you without the hassle of the terminal, but I'm going to highlight four of them here, one that I personally use (and I promise I'm not being paid to advertise it), and three that many people in the community use and I've heard nothing but good things about; they are LiveReload, CodeKit, Scout App, and Compass.app.

LiveReload is a nifty little server app that, at its core, listens for changes in a folder (like a change in a Sass file in your theme) and refreshes the tab you are in with the new changes, allowing for a great Design In Browser workflow. LiveReload is available for both Windows and Mac computers and will compile not only Sass+Compass but Less, Stylus, CoffeeScript, IcedCoffeeScript, Eco, SLIM, HAML, and Jade as well. I personally use LiveReload for its Design in Browser capabilities and compile my CSS through the terminal, so its flexible like that. LiveReload uses your installed Ruby gems, making it fully integrated with all of the tools I've talked about here. You can also use LiveReload without having it compile your files for you, just for reloading your browser.

CodeKit is very similar to LiveReload in that it will compile your files for you, reload your browser, etc… It is only available for Mac computers, but is very pretty and has a nicer user interface than LiveReload.

Scout App is a free cross-platform application for Windows and Mac computers that has a self-contained environment to run Sass+Compass and compile your CSS for you, but it does not have the browser refreshing like LiveReload has.

Compass.app is the most widely supported Application compiler, working on all three major platforms. It is similar to Scout, but includes LiveReload integration for browser reloading.