1. Introducing Less, "Leaner CSS"

Topic

Expanding upon the new thread of Web development knowledge we started last week, we look this week at the CSS pre-processing language known as Less, short for "Leaner CSS" according its original authors.

What is Less and what does it do?

As the authors of Less put it:

Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themeable and extendable.

Less is an ongoing language project. The project's online community site can be found at **http://lesscss.org/** **(http://lesscss.org/)** .

CSS Pre-Processing Languages

We have seen how the HTML code of websites today is not composed simply of static text files, but instead is dynamically generated in systems like WordPress that use PHP and MySQL database storage. Now let us turn our attention to the world of dynamically-generated CSS code. Developers have found challenges writing CSS rules for a number of reasons over the years. CSS has no ability to define variables for things like color and size, so we must use repeated literal values over and over again. CSS rules cannot be nested, but HTML can, so CSS provides selector types like descendent selectors, but no true nesting to indicate selector targets. Two projects have grown in response to the desire to improve the CSS language: Less and Sass. The Bootstrap version 3.x framework is created using Less source files for the CSS rules in the framework. *Note that in version 4.x of Bootstrap, that project switches to use Sass source files for the generation of its CSS rules.*

Both Less and Sass generate one or more compiled CSS files from a group of source files. Both pre-processing languages extend CSS to allow the use of variables, nested rules, and mixins – or reusable rules. When you make a change to a Less or Sass source file, a compiler is run to make the final CSS files included in your project. In this way, CSS rule generation still yields a static CSS file, but one that is regenerated every time a source file changes.

Basic Characteristics of CSS Pre-Processors

1. Dynamically generate CSS file(s) after source file change.
2. Add capabilities unavailable in CSS language.
3. CSS compilers available in a number of languages, including JavaScript and PHP.

The WP-LESS plug-in

The WordPress platform we are using allows additional server-side code installations, using PHP files known as plug-ins. Last week you installed the WP-LESS plug-in and activated it for your site. The PHP code in this plug-in is responsible for generating the final CSS files for your Bootstrap-powered WordPress theme.

Special Topic 4: The Less CSS Pre-Processor

Participation in the **"CSS Pre-Processors" Discussion (https://santarosajc.instructure.com/courses/17705/discussion_topics/64531)** is required this week. It is worth 15 points of your total class point score. A full score of 15 points can be earned by posting two or more times before this module closes. Make your first post with your initial thoughts and any questions after doing all of the following tasks:

1. Read the Less Features Guide found at **http://lesscss.org/features/** **(http://lesscss.org/features/)** .
2. Make some changes to the Bootstrap framework Less files in your WordPress-powered site's custom theme.

Follow up with a second post responding to your classmates' shared perspectives on web content management and WordPress..

Objectives

By the end of this week you will be able to...

- Learn the basic concepts of a CSS Pre-Processor such as Less.
- Work with Less code to generate CSS for your WordPress-powered theme.
- Develop an understanding and perspective about CSS Pre-Processors and Web Content Management Systems in dialog with classmates and instructor.

Note: You will not have to learn the PHP server-side scripting language – all PHP code is provided.