COMP 1950

Web Development and Design 2

Day 08

Agenda

- Sass

About Sass

- Sass stands for:

S yntactically

A wesome

S tyle

S heets

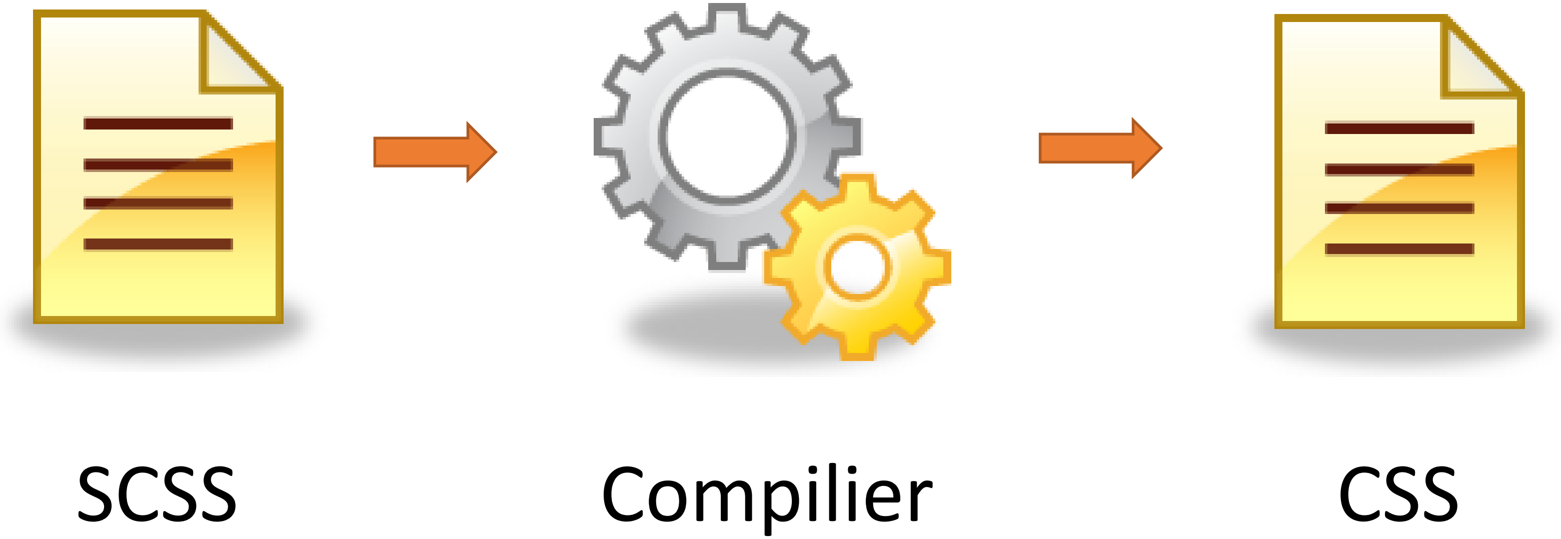
About Sass

- Sass is a CSS preprocessor and scripting language
- Developers write CSS like syntax that is then interpreted by the Sass complier and converted into regular CSS
- Sass is open sourced and coded in Ruby; however other implementations exist.
- Developers can have script like abilities for their CSS development. The following are available to CSS developers that use Sass:
    - Variables
    - Mixins (sort of like functions)
    - Functions
    - If/else statements
    - Loops
    - @extend
    - Importing and concatenation of several files into a single file
    - Compression
    - “//” single line comments

Why Use Sass

- It adds programmability to CSS
- It allows less repeated code in your CSS development (before compiling)
  - You can use variables, mixins, functions and extends to write a set of rules once and apply them in multiple locations
- It allows you to break up your CSS into smaller modular files and concatenate them into a single CSS file, making development easier
- Promotes DRY (Do not repeat yourself) principals in writing your CSS
- We can use the quick and easy “//” for single line comments

How Sass Works



SCSS

Compiler

CSS

Installing Sass

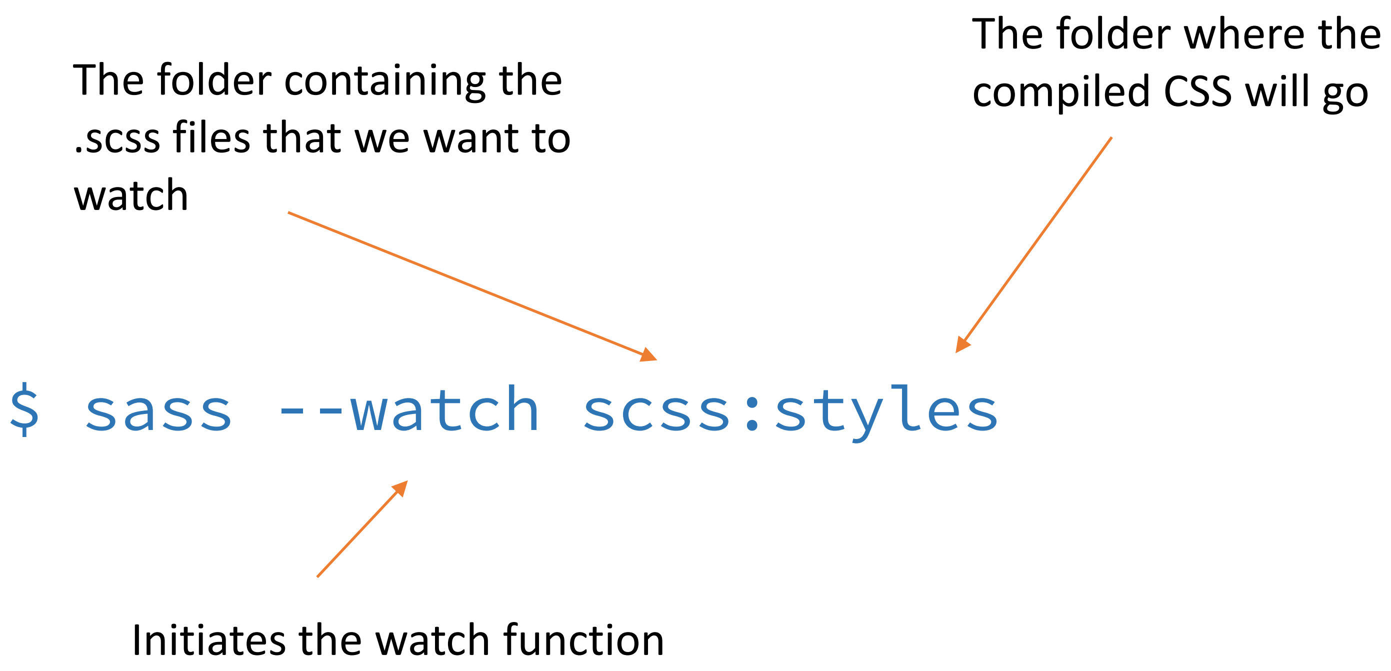
- Many options to get Sass up and running on your computer
  - Use a build system such as Gulp
  - Use a GUI tool such as CodeKit or Prepros
  - Install Ruby, install Sass and run from the Terminal (Mac) or Command Prompt (Windows)
- For this class we will run Sass from Terminal (Mac) or the Command Prompt (Windows)

Installing Sass

- To get Sass up running using the Ruby version of Sass you will need to install Ruby on your system
    - Mac’s already have Ruby installed
    - Linux and Windows users will need to install Ruby on their systems
        - Windows users can visit the link below for an easy Ruby installer that installs Ruby and gives you a Ruby command prompt program
            - http://www.rubyinstaller.org
- Once Ruby is installed you can open the terminal on a Mac or Linux machine or the Ruby command prompt on Windows and type:
    - $ gem install sass
        - Mac users may have to type "sudo gem install sass"
            - Don't want to use "sudo"?
                - Follow the instructions laid out in the "mac-os-sass-install-instructions.pdf" file located in the day's folder
    - Sass will download and install
- To double check if Sass has installed correctly type:
    - $ sass –v
    - You should see something similar displayed in the terminal:
        - Sass 3.3.4 (Maptastic Maple)

Watch Files With Sass

- In order for Sass to compile files you need to tell Sass what folder to watch
- To tell Sass to watch a file or folder of files first navigate to your project folder in the terminal and then type the following:



The folder containing the .scss files that we want to watch

The folder where the compiled CSS will go

```
$ sass --watch scss:styles
```

Initiates the watch function

Your Browser only understands CSS

- A key point to remember is that your browser only reads CSS files, so the link to your stylesheet should be to your compiled CSS file, not to your Sass files
- The Sass compiler will create a ".map" file which some browser's developer tools can read. The ".map" file will tell the browser which Sass file and line number a particular CSS rule was compiled from
    - That information is displayed in the browser's developer tools to aid in CSS/Sass development

Determining CSS Output Format

- Sass can output the CSS in a variety of ways, the default is using the nested syntax where nested rules are indented
  - Other options include:
    - expanded
      - Each selector written on a new line with each property and value also written on its own line (this is my personal preference as it is how I write my CSS)
    - compact
      - Each selector on its own line and all properties and values written on the same line as the selector
    - compressed
      - All non-required white space between selectors, properties and values are removed
        - This outputs the smallest file size

Determining CSS Output Format

- To output CSS with the expanded output type the following into the terminal:

Tells Sass to output CSS using expanded CSS syntax

```
$ sass --watch --style expanded scss:styles
```

Don't Edit the Compiled CSS File

- Do not edit the compiled CSS
- If you need to change the CSS, write your changes in Sass and re-compile
- Feel free to look at the compiled CSS to see what Sass is outputting and for troubleshooting purposes

Writing Sass

- Sass can be written in two ways
  - Sass syntax
    - Uses indenting to format rules
  - SCSS syntax
    - Written using a syntax that is similar to the syntax used for writing standard CSS
    - For this course we will be using SCSS syntax but your choice of syntax comes down to either personal preference or what your development team is using

Sass Syntax

```
$brand-color: #eee;

p
  font-size: 24px
  color: $brand-color
  line-height: 1.5
```

SCSS Syntax

```
$brand-color: #eee;

p {
  font-size: 24px;
  color: $brand-color;
  line-height: 1.5;
}
```

Sass Resources

- Sass web site
  - http://sass-lang.com
- Sass book
  - Sass for Web Designers
    - http://www.abookapart.com/products/sass-for-web-designers

Image credit: http://www.abookapart.com/products/sass-for-web-designers

VS Code – Sass Extension

- Useful extension for Sass in Visual Studio Code
  - SCSS IntelliSense
    - https://marketplace.visualstudio.com/items?itemName=mrmlnc.vscode-scss
    - This extension will auto-complete variables, mixins, functions and extends for a values located across the project
      - Especially useful when using partials