Intro to Sass

Women Who Code Workshop

Fiona Chan
@mobywhale

Cathy Lill
@cathyblabla

Who's had experience with Sass/CSS pre-processor before?

What we're covering tonight

6:30pm	Getting started
	Syntax and features
	Debugging
7:15pm-ish	Break
7:30pm	Your turn

Sass = CSS pre-processor

Not SASS

Extends CSS with features

(nested selectors, variables, functions)

Sass does not help you write better CSS

Install Sass

Install Sass

- GUI:
  http://koala-app.com/

- Command line:
  http://sass-lang.com/install

Basic syntax

Scss

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

http://sass-lang.com/guide

Comment

```
1  // This is a Scss comment
2  // Second line of my amazing comment
```

@import partials

Without Sass, one giant CSS!!!

```
5593  .visible-print {
5594    display: none !important;
5595  }
5596  tr.visible-print {
5597    display: none !important;
5598  }
5599  th.visible-print,
5600  td.visible-print {
5601    display: none !important;
5602  }
5603  @media print {
5604    .visible-print {
5605      display: block !important;
5606    }
5607    tr.visible-print {
5608      display: table-row !important;
5609    }
5610    th.visible-print,
5611    td.visible-print {
5612      display: table-cell !important;
5613    }
5614    .hidden-print {
5615      display: none !important;
5616    }
5617    tr.hidden-print {
5618      display: none !important;
5619    }
5620    th.hidden-print,
5621    td.hidden-print {
5622      display: none !important;
5623    }
5624  }
5625  .fade {
5626    opacity: 0;
5627    -webkit-transition: opacity 0.15s linear;
5628    transition: opacity 0.15s linear;
5629  }
5630  .fade.in {
5631    opacity: 1;
5632  }
5633  .collapse {
5634    display: none;
5635  }
5636  .collapse.in {
5637    display: block;
5638  }
5639  .collapsing {
5640    position: relative;
5641    height: 0;
5642    overflow: hidden;
5643    -webkit-transition: height 0.35s ease;
5644    transition: height 0.35s ease;
5645  }
```

Line 5645

Partials let you split up your CSS into smaller, more manageable chunks

Sass compiles all the partials into a single CSS file in the end

File naming

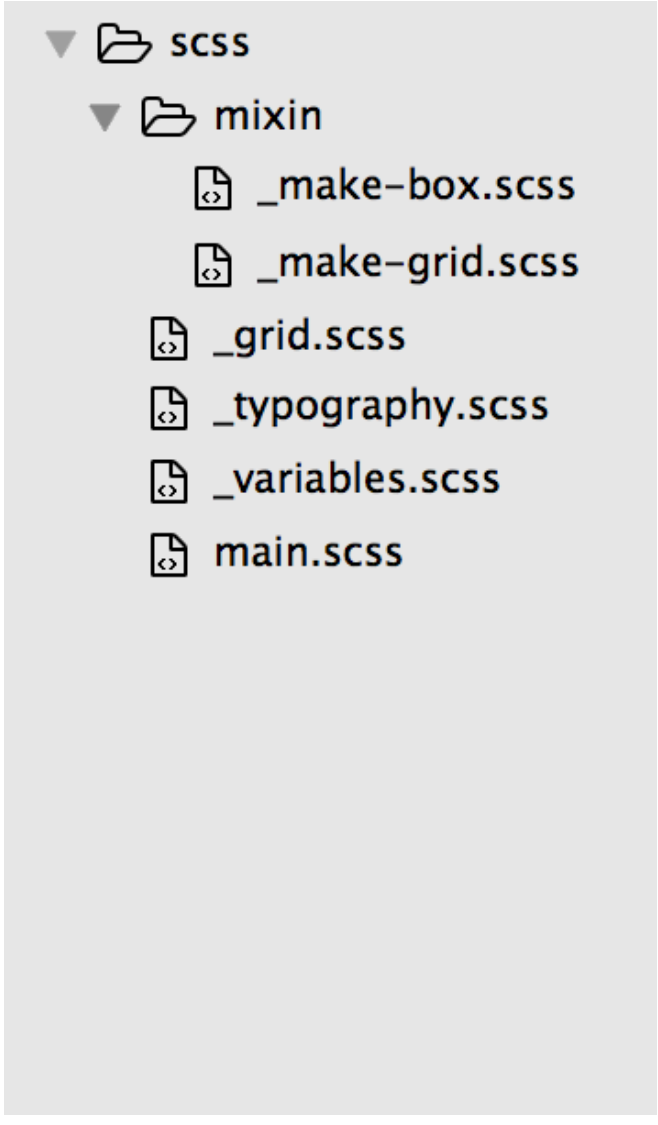
- `my-file.scss`

  Sass will output `my-file.css`

- `_my-partial.scss`

  File name prepended with underscore, Sass won’t output any CSS file

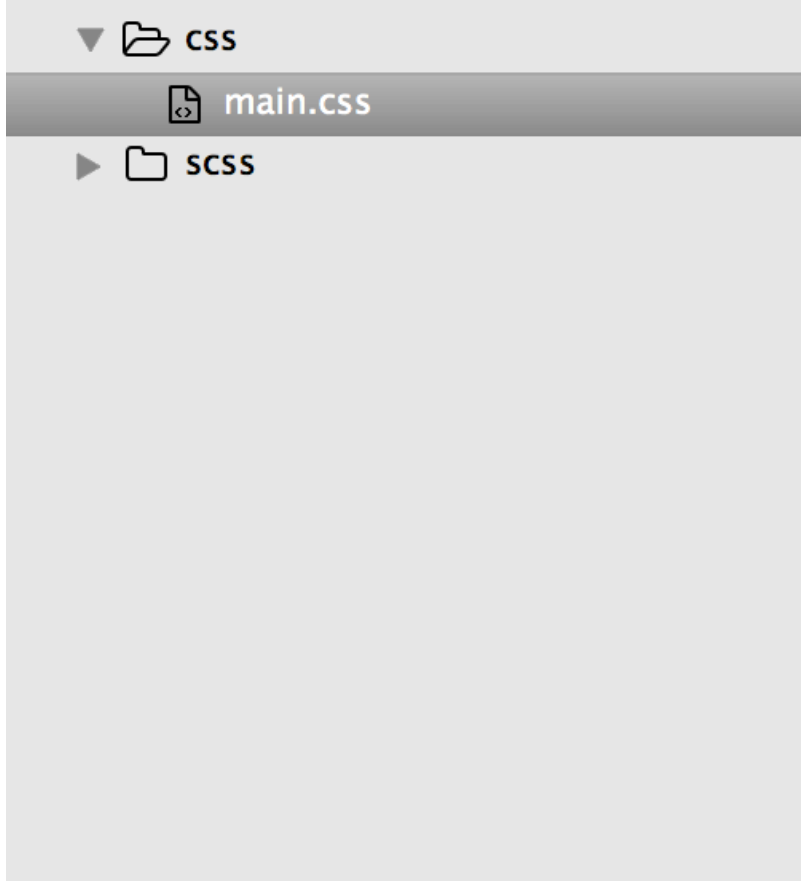
@import partials




```
//main.scss

@import 'variables';
@import 'grid';
@import 'typography';
```

`@import` partials

Outputs `main.css`



$Variables

Store information you can re-use throughout your stylesheet

http://sass-lang.com/guide

Scss

```scss
$grey: #666;

h1 {
  background: $grey;
  margin: 20px 0;
}

p {
  color: $grey;
}
```

CSS

```css
h1 {
  background: #666;
  margin: 20px 0;
}

p {
  color: #666;
}
```

Scss

```scss
$grey: #666;
$grey: #999;

h1 {
  background: $grey;
  margin: 20px 0;
}

p {
  color: $grey;
}
```

CSS

```css
h1 {
  background: #999;
  margin: 20px 0;
}

p {
  color: #999;
}
```

You must
declare your variables first
before they can be used

Nesting

Nest selectors following the hierarchy of your DOM elements

HTML

```
<ul class="nav">
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2</a></li>
    <li><a href="#">Link 3</a></li>
</ul>
```

Scss

```
.nav {
  li {
    margin: 5px;

    a {
      color: #BADA55;
    }
  }
}
```

CSS

```
.nav li {
  margin: 5px;
}
.nav li a {
  color: #BADA55;
}
```

Scss

```
.nav {
  li {
    margin: 5px;

    a {
      color: #BADA55;
    }
  }
}
```

HTML

```
<ul class="nav">
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2</a></li>
    <li class="active">
        <a href="#">Link 3</a>
    </li>
</ul>
```

Scss

```
.nav {
  li {
    margin: 5px;

    a {
      color: blue;
    }
  }

  li.active {
    a {
      color: pink;
    }
  }
}
```

& Parent selector

HTML

```
<ul class="nav">
    <li><a href="#">Link 1</a></li>
    <li><a href="#">Link 2</a></li>
    <li class="active">
        <a href="#">Link 3</a>
    </li>
</ul>
```

Scss

```
.nav {
  li {
    margin: 5px;

    a {
      color: blue;
    }

    &.active {
      a {
        color: pink;
      }
    }
  }
}
```

CSS

```css
.nav li {
  margin: 5px;
}
.nav li a {
  color: blue;
}
.nav li.active a {
  color: pink;
}
```

Scss

```scss
.nav {
  li {
    margin: 5px;

    a {
      color: blue;
    }

    &.active {
      a {
        color: pink;
      }
    }
  }
}
```

Scss

```scss
.nav {
  li {
    margin: 5px;

    a {
      color: blue;

      &:hover,
      &:focus {
        color: salmon;
      }
    }
  }
}
```

CSS

```css
.nav li {
  margin: 5px;
}
.nav li a {
  color: blue;
}
.nav li a:hover, .nav li a:focus
{
  color: salmon;
}
```

Nesting is cool!

Scss

```scss
body {
  header {
    background: coffee;

    ul {
      li {
        margin: 5px;

        a {
          color: salmon;
        }
      }
    }
  }
}
```

CSS

```css
body header {
  background: coffee;
}
body header ul li {
  margin: 5px;
}
body header ul li a {
  color: salmon;
}
```

Unnecessarily long selector

Don’t go nesting ALL the selectors!!!

Long selectors = High specificity = PAIN!

Scss

```
header {
  background: coffee;
}

.header-nav {
  li {
    margin: 5px;
  }

  a {
    color: salmon;
  }
}
```

CSS

```
header {
  background: coffee;
}

.header-nav li {
  margin: 5px;
}
.header-nav a {
  color: salmon;
}
```

@extend

Allows a selector to
inherit styles of another one

Useful when you want to group common styles together

Box

Box plain

Box fancy

```
.box {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;

  border: 1px solid #000;
}

.box-fancy {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;

  box-shadow: 0 0 5px lime;
}
```

Scss

```scss
.box {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  @extend .box;
  border: 1px solid #000;
}

.box-fancy {
  @extend .box;
  box-shadow: 0 0 5px lime;
}
```

CSS

```css
.box, .box-plain, .box-fancy {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  border: 1px solid #000;
}

.box-fancy {
  box-shadow: 0 0 5px lime;
}
```

Beware: `@extend` will extend all instances of that selector

Scss

```scss
.box {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  @extend .box;
  border: 1px solid #000;
}

.box-fancy {
  @extend .box;
  box-shadow: 0 0 5px lime;
}

aside .panel .box {
  background: purple;
}

header .box {
  background: blue;
}
```

CSS

```
.box, .box-plain, .box-fancy {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  border: 1px solid #000;
}

.box-fancy {
  box-shadow: 0 0 5px lime;
}

aside .panel .box, aside .panel .box-plain, aside .
.box-fancy {
  background: purple;
}

header .box, header .box-plain, header .box-fancy {
  background: blue;
}
```

%Placeholder selector

Scss

```scss
%box {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  @extend %box;
  border: 1px solid #000;
}

.box-fancy {
  @extend %box;
  box-shadow: 0 0 5px lime;
}

aside .panel .box {
  @extend %box;
  background: purple;
}

header .box {
  @extend %box;
  background: blue;
}
```

CSS

```css
.box-plain, .box-fancy, aside .panel .box,
header .box {
  color: orange;
  padding: 20px;
  background: #fff;
  margin-bottom: 20px;
}

.box-plain {
  border: 1px solid #000;
}

.box-fancy {
  box-shadow: 0 0 5px lime;
}

aside .panel .box {
  background: purple;
}

header .box {
  background: blue;
}
```

Mixin

An abstraction of a common pattern into reusable snippets

Scss

```scss
@mixin clearfix() {
  &:before,
  &:after {
    content: "";
    display: table;
  }

  &:after {
    clear: both;
  }
}

.row {
  @include clearfix();
}

.nav {
  @include clearfix();
}
```

CSS

```css
.row:before, .row:after {
  content: "";
  display: table;
}
.row:after {
  clear: both;
}

.nav:before, .nav:after {
  content: "";
  display: table;
}
.nav:after {
  clear: both;
}
```

Can also pass in arguments

Scss

```scss
@mixin rotate($deg) {
  -webkit-transform: rotate($deg);
  -moz-transform: rotate($deg);
  transform: rotate($deg);
}

.box {
  @include rotate(45deg);
}
```

CSS

```css
.box {
  -webkit-transform: rotate(45deg);
  -moz-transform: rotate(45deg);
  transform: rotate(45deg);
}
```

Scss

```scss
@mixin breakpoint($size) {
  @if $size == small {
    @media (min-width: 20em) { @content; }
  }
  @else if $size == medium {
    @media (min-width: 37.5em) { @content; }
  }
  @else if $size == large {
    @media (min-width: 65.25em) { @content; }
  }
}

header {
  @include breakpoint(medium) {width: 50%;}
}
```

CSS

```css
@media (min-width: 37.5em) {
  header {
    width: 50%;
  }
}
```

Existing library

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

- Bourbon
  http://bourbon.io/

Compass or Bourbon?

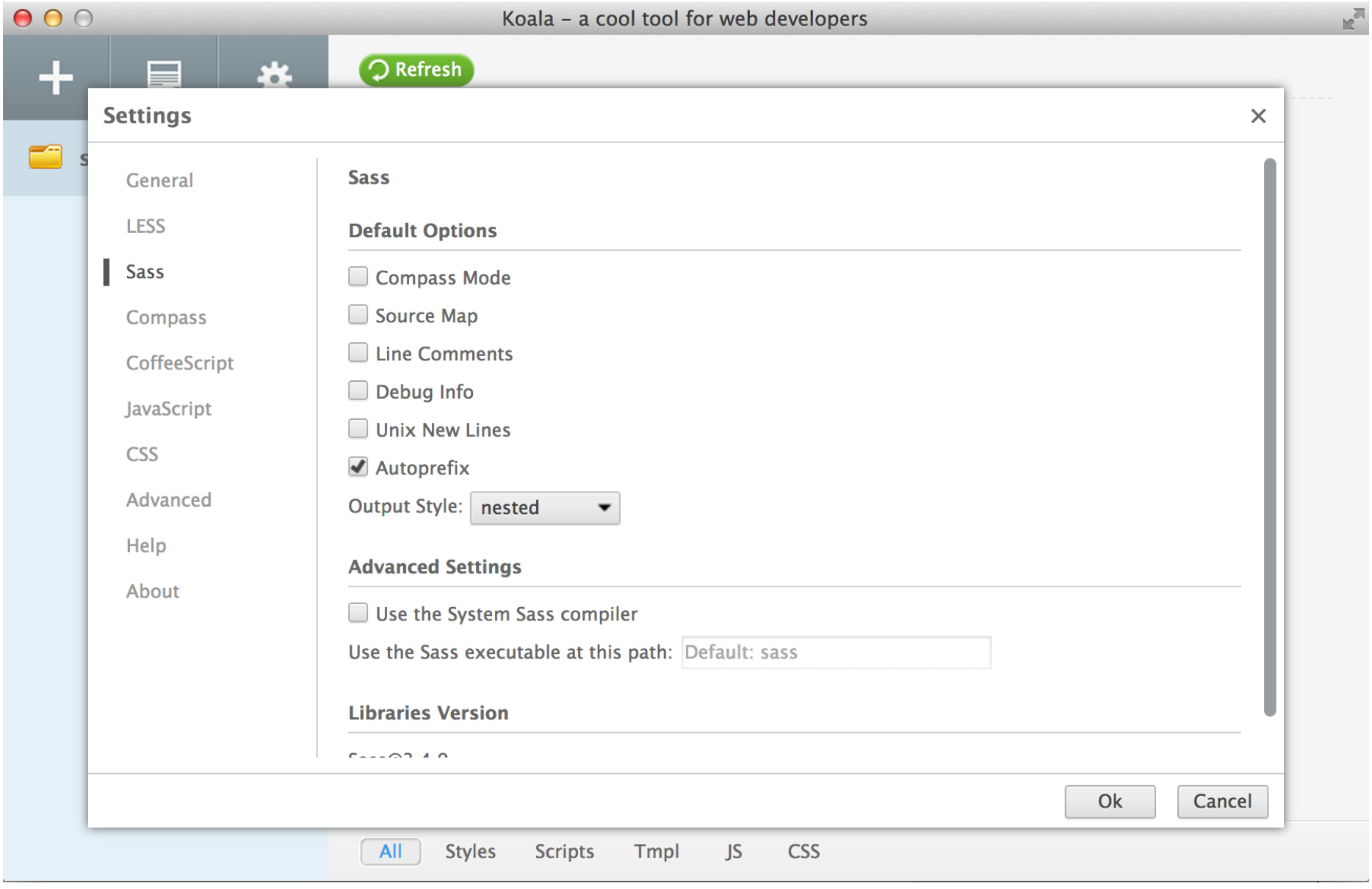
	Compass	Bourbon
Prefixes	yep (from *Can I Use…*)	yep
Typography	good	decent (good with Bitters)
Grid	nope	with Neat
Helpers	yep	yep + Bitters
Sprites	yep	nope
Community	excellent	very good
Weight	heavy	lightweight

http://www.sitepoint.com/compass-or-bourbon-sass-frameworks/

Just vendor prefixing?
Check out Autoprefixer

https://github.com/postcss/autoprefixer

Easy peasy in Koala!



Mixin

vs

Placeholder

Remember this example?

```scss
@mixin clearfix() {
  &:before,
  &:after {
    content: "";
    display: table;
  }

  &:after {
    clear: both;
  }
}

.row {
  @include clearfix();
}

.nav {
  @include clearfix();
}
```

```scss
.row:before, .row:after {
  content: "";
  display: table;
}
.row:after {
  clear: both;
}

.nav:before, .nav:after {
  content: "";
  display: table;
}
.nav:after {
  clear: both;
}
```

Let’s try the same thing with placeholder selector

Scss

```scss
%clearfix {
  &:before,
  &:after {
    content: "";
    display: table;
  }

  &:after {
    clear: both;
  }
}

.row {
  @extend %clearfix;
}

.nav {
  @extend %clearfix;
}
```

CSS

```css
.row:before, .nav:before, .row:after,
.nav:after {
  content: "";
  display: table;
}
.row:after, .nav:after {
  clear: both;
}
```

Using mixin

```
.row:before, .row:after {
  content: "";
  display: table;
}
.row:after {
  clear: both;
}

.nav:before, .nav:after {
  content: "";
  display: table;
}
.nav:after {
  clear: both;
}
```

Using placeholder

```
.row:before, .nav:before, .row:after,
.nav:after {
  content: "";
  display: table;
}
.row:after, .nav:after {
  clear: both;
}
```

@function

Similar to a mixin, but the output from a function is a single value

- numbers, strings, colors, booleans, or lists

http://thesassway.com/advanced/pure-sass-functions

Scss

```scss
@function percentage($target, $container) {
  @return (($target / $container) * 100%)
}

.main {
  width: percentage(800px, 1000px);
}
```

CSS

```css
.main {
  width: 80%;
}
```

Built-in Sass functions

```
lighten($color, $amount);
darken($color, $amount);
saturate($color, $amount);
invert($color, $amount);
```

http://sass-lang.com/documentation/Sass/Script/Functions.html

Sass control directives

```
@if

@for

@each

@while
```

Debugging

@debug

Scss

```
$variable: blue;
$variable: red;

@debug $variable;
```

Console

```
>>> Change detected at 23:10:00 to: screen.scss
/Users/fionachan/Documents/Work/Sass workshop/sass-
workshop/sass/screen.scss:11 DEBUG: red
```

Source map

Shows you which line in the pre-compiled Sass the CSS is from

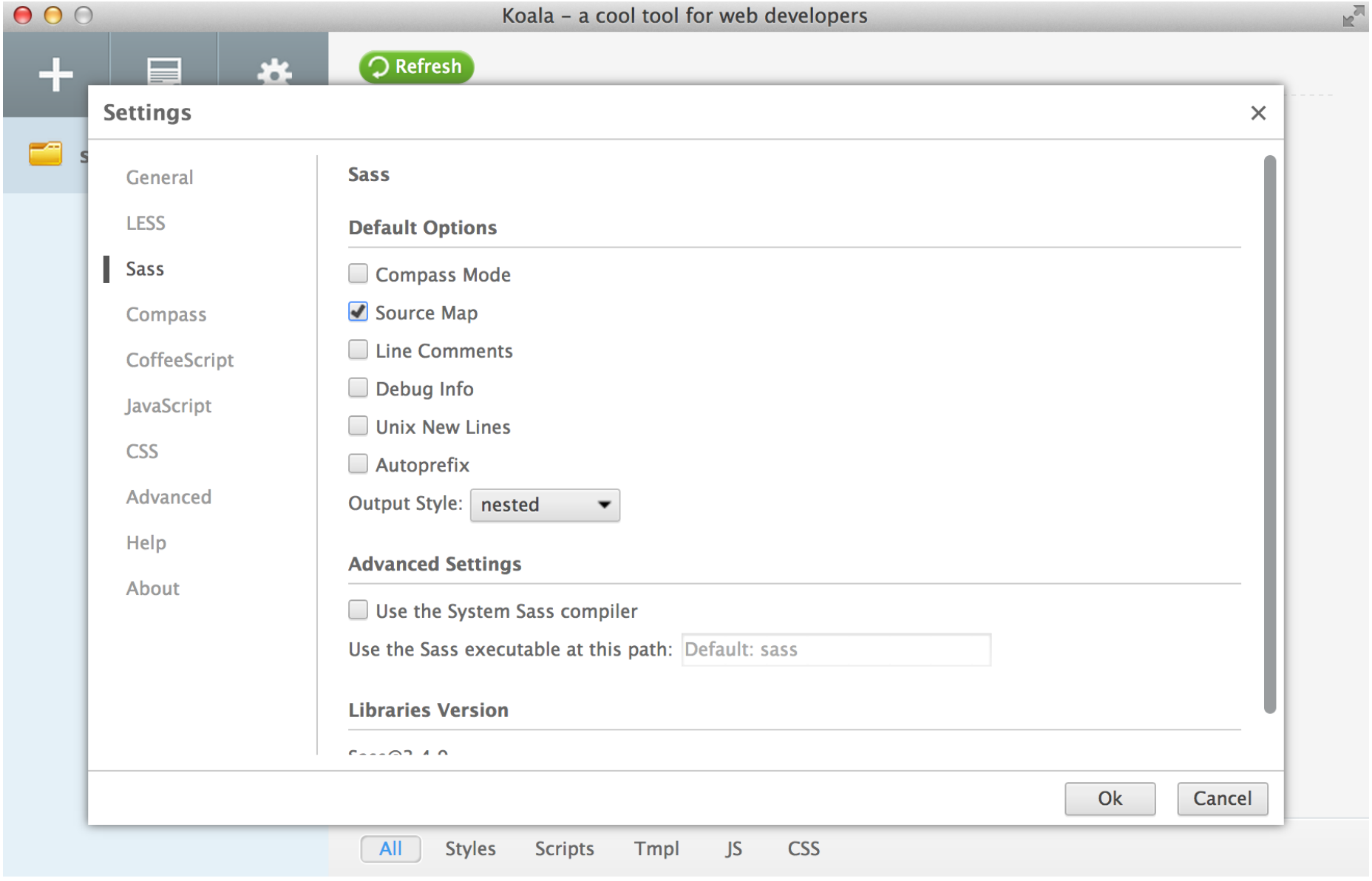
Elements | Network | Sources | Timeline | Profiles | Resources | Audits | Console | SnappySnippet

```
<!DOCTYPE html>
▼<html lang="en">
  ▶<head>…</head>
  <!--
        Hello! Thank you for checking out
    -->
▼<body data-pinterest-extension-installe
  ▼<div class="content">
    ▼<div class="container">
      ▶<header role="banner">…</header>
      ▼<main role="main">
          <!-- Show last 5 posts here -->
        ▼<div class="articles">
          ▶<article>…</article>
          ▼<article>
            ▼<header>
                <time class="type-muted" d
              ▼<h2>
                  <a href="/2014/09/its-not
                </h2>
              </header>
            ▶<p>…</p>
            ▶<p>…</p>
```

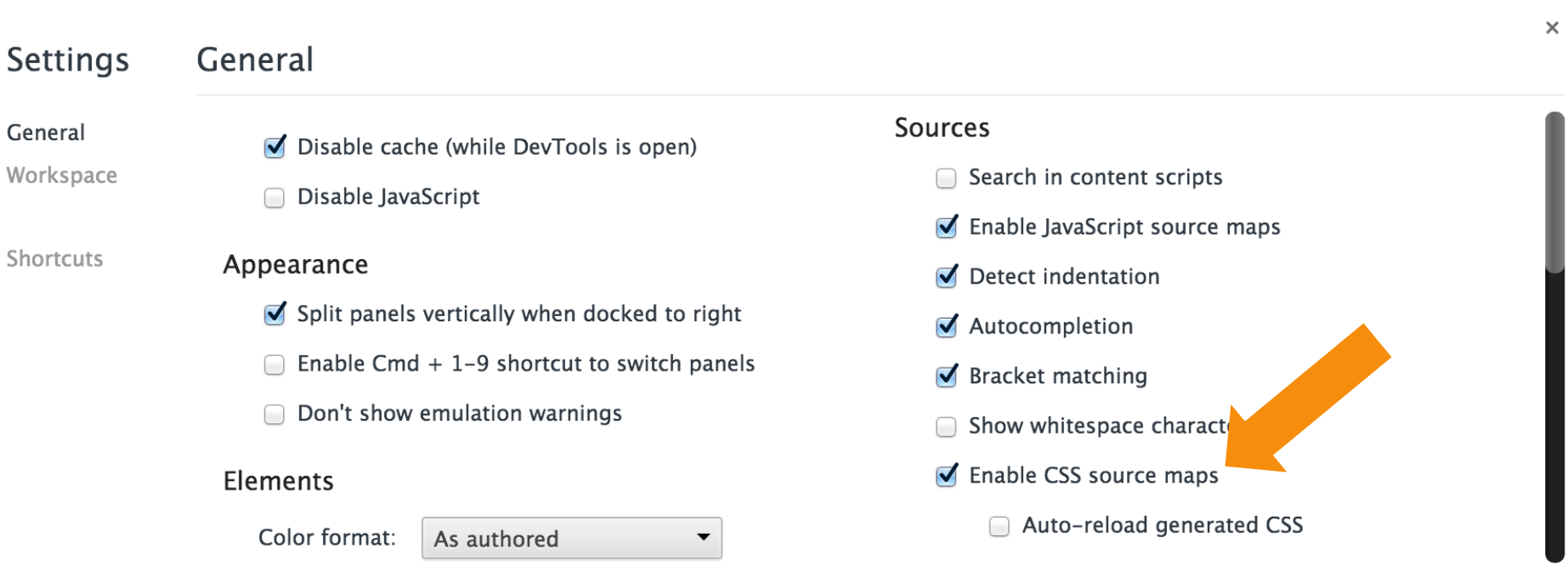
Styles | Computed | Event Listeners | DOM Breakpoints | Proper

```
element.style {
}

html {                                                    _template.scss:4
    height: 100%;
    padding-top: 15px;
    background-color: #b8e9eb;
    background-image: linear-gradient(-45deg, #7cd7da 25%, transparent
      25%, transparent 50%, #7cd7da 50%, #7cd7da 75%, transparent
      75%, transparent);
    background-size: 40px 40px;
}

html, button, input, select, textarea {                   _normalize.scss:109
    font-family: sans-serif;
}

html {                                                    _normalize.scss:97
    background: ▶ #fff;
    color: #000;
⚠   -webkit-text-size-adjust: 100%;
    -ms-text-size-adjust: 100%;
}
```

Koala makes it so easy



Enabled by default in Chrome Developer Tool



Enable Source map in command line:

Compiling with Sass

```
sass sass/screen.scss:stylesheets/screen.css --sourcemap
```

http://thesassway.com/intermediate/using-source-maps-with-sass

Few rules to remember

1. Inception rule

Don’t nest more than 3 levels deep

2. Always check your output CSS

Otherwise...

```
.lrn_widget.lrn_association .lrn_btn_drag.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_association .lrn_btn_drag.lrn_valid[disabled]:after,
.lrn_widget.lrn_association .lrn_btn_drag.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_association .lrn_btn_drag.lrn_correct[disabled]:after,
.lrn_widget.lrn_association .lrn_btn_drag.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_association .lrn_btn_drag.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_association .lrn_btn_sort.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_association .lrn_btn_sort.lrn_valid[disabled]:after,
.lrn_widget.lrn_association .lrn_btn_sort.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_association .lrn_btn_sort.lrn_correct[disabled]:after,
.lrn_widget.lrn_association .lrn_btn_sort.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_association .lrn_btn_sort.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_drag.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_drag.lrn_valid[disabled]:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_drag.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_drag.lrn_correct[disabled]:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_drag.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_drag.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_sort.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_sort.lrn_valid[disabled]:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_sort.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_sort.lrn_correct[disabled]:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_sort.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_clozeassociation .lrn_btn_sort.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_drag.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_drag.lrn_valid[disabled]:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_drag.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_drag.lrn_correct[disabled]:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_drag.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_drag.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_sort.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_sort.lrn_valid[disabled]:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_sort.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_sort.lrn_correct[disabled]:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_sort.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_imageclozeassociation .lrn_btn_sort.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_orderlist .lrn_btn_drag.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_orderlist .lrn_btn_drag.lrn_valid[disabled]:after,
.lrn_widget.lrn_orderlist .lrn_btn_drag.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_orderlist .lrn_btn_drag.lrn_correct[disabled]:after,
.lrn_widget.lrn_orderlist .lrn_btn_drag.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_orderlist .lrn_btn_drag.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_orderlist .lrn_btn_sort.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_orderlist .lrn_btn_sort.lrn_valid[disabled]:after,
.lrn_widget.lrn_orderlist .lrn_btn_sort.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_orderlist .lrn_btn_sort.lrn_correct[disabled]:after,
.lrn_widget.lrn_orderlist .lrn_btn_sort.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_orderlist .lrn_btn_sort.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_classification .lrn_btn_drag.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_classification .lrn_btn_drag.lrn_valid[disabled]:after,
.lrn_widget.lrn_classification .lrn_btn_drag.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_classification .lrn_btn_drag.lrn_correct[disabled]:after,
.lrn_widget.lrn_classification .lrn_btn_drag.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_classification .lrn_btn_drag.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_classification .lrn_btn_sort.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_classification .lrn_btn_sort.lrn_valid[disabled]:after,
.lrn_widget.lrn_classification .lrn_btn_sort.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_classification .lrn_btn_sort.lrn_correct[disabled]:after,
.lrn_widget.lrn_classification .lrn_btn_sort.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_classification .lrn_btn_sort.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_sortlist .lrn_btn_drag.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_sortlist .lrn_btn_drag.lrn_valid[disabled]:after,
.lrn_widget.lrn_sortlist .lrn_btn_drag.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_sortlist .lrn_btn_drag.lrn_correct[disabled]:after,
.lrn_widget.lrn_sortlist .lrn_btn_drag.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_sortlist .lrn_btn_drag.lrn_incorrect[disabled]:after,
.lrn_widget.lrn_sortlist .lrn_btn_sort.lrn_valid.lrn_disabled:after,
.lrn_widget.lrn_sortlist .lrn_btn_sort.lrn_valid[disabled]:after,
.lrn_widget.lrn_sortlist .lrn_btn_sort.lrn_correct.lrn_disabled:after,
.lrn_widget.lrn_sortlist .lrn_btn_sort.lrn_correct[disabled]:after,
.lrn_widget.lrn_sortlist .lrn_btn_sort.lrn_incorrect.lrn_disabled:after,
.lrn_widget.lrn_sortlist .lrn_btn_sort.lrn_incorrect[disabled]:after {
    filter: progid:DXImageTransform.Microsoft.Alpha(Opacity=100); opacity: 1;
}
```

3. Keep it simple

You don’t have to use all the power Sass gives you

4. You're still writing CSS at the end of the day

Sass is just a tool to help you

Resources

- Official Sass doc

  http://sass-lang.com

- Great beginners - advanced Sass articles

  http://thesassway.com/

- A set of Sass guidelines

  http://sass-guidelin.es/

Thank you for listening :)