

CSS PRE-PROCESSORS

Sass / Less / Stylus

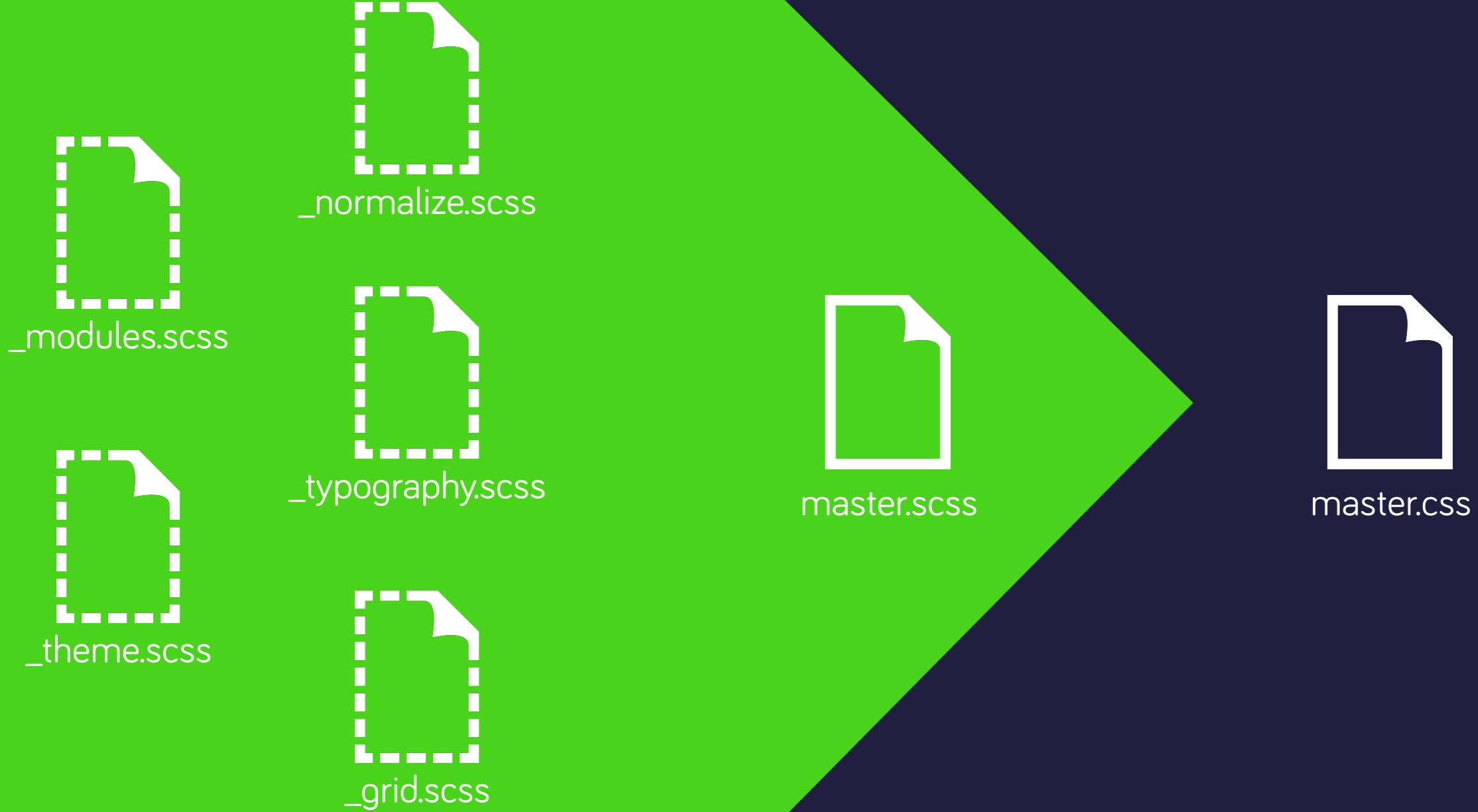
TOPICS

1. What is a preprocessor
2. Preprocessors (Less, Sass, Stylus)
3. Setup
4. Features (beginner, intermediate, advanced)
5. Frameworks

CSS PRE-PROCESSORS

What is a Preprocessor?







Expressive, dynamic, robust CSS

CSS needs a hero

```
body {  
    font: 12px Helvetica, Arial, sans-serif;  
}  
a.button {  
    -webkit-border-radius: 5px;  
    -moz-border-radius: 5px;  
    border-radius: 5px;  
}
```

Selectors
Variables
Interpolation
Operators
Mixins
Functions
Keyword Arguments
Built-in Functions
Rest Params
Comments
Conditionals
Hashes
Iteration
@import and @require
@media
@font-face
@keyframes
@extend
@block
url()
CSS Literal
CSS Style Syntax
Char Escaping
Executable
Error Reporting
Connect Middleware
Introspection API
JavaScript API
CSS3 Extensions with Nib
Try Stylus Online!

What if we could omit braces?

```
body  
font: 12px Helvetica, Arial, sans-serif;
```

Getting started

An overview of Less, how to download and use, examples and more.

less v1.6.3 has been released - [See what's new](#)

[Getting Started](#)

[Using Less](#)

[Client-side usage](#)

[License FAQs](#)

Getting Started

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, themable and extendable.

Less runs inside Node, in the browser and inside Rhino. There are also many 3rd party tools that allow you to compile your files and watch for changes.

For example:

```
@base: #f938ab;  
  
.box-shadow(@style, @c) when (iscolor(@c)) {  
  -webkit-box-shadow: @style @c;
```

[Install](#)[Learn Sass](#)[Documentation](#)[Get Involved](#)

CSS with superpowers



Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

Current Release: Media Mark (3.2.12)

[Release Notes](#)[Fork on Github](#)

CSS COMPATIBLE

Sass is completely compatible with all versions of CSS. We take this compatibility seriously, so that you can seamlessly use any available CSS libraries.

FEATURE RICH

Sass boasts more features and abilities than any other CSS extension language out there. The Sass Core Team has worked endlessly to not only keep up, but stay ahead.

MATURE

Sass has been actively supported for almost 7 years by its loving Core Team.

CSS PRE-PROCESSORS

*If you write sucky CSS,
a pre-processor won't
make it suck less.*

QUICK START

GUI APPS



CodeKit

Files Log

SEARCH

PROJECTS

All Files Styles Scripts Pages Images

Crazy Client /index.php

My Website /scripts/jquery.js
This file does not directly compile.

Steve /scripts/pageController.js

UNC /scripts/min/pageController-min.js

FRAMEWORKS

VDKFramework /scripts/storeController.coffee

Twitter Bootstrap /scripts/min/storeController.js

haml /store/storePage.haml

/store/storePage.html

sass /styles/about.sass

/css/about.css

less /styles/main.less

/css/main.css

styl /styles/store.styl

/css/store.css

Javascript

Do not compile directly

Check Syntax With:

JSHint

When Processing:

Concatenate, then minify

Check Syntax

Imports:

jquery.js Prepend

Imported By:

Process

A screenshot of the CodeKit application window. The left sidebar lists projects like 'Crazy Client' and 'My Website'. The main pane shows files like '/index.php', '/scripts/jquery.js', and '/scripts/pageController.js'. A right-hand panel for 'Javascript' includes settings for compilation, syntax checking (set to JSHint), and processing (set to 'Concatenate, then minify'). It also shows an 'Imports' section with 'jquery.js' listed and a 'Prepend' dropdown, and an 'Imported By' section which is currently empty. Buttons for 'Check Syntax' and 'Process' are at the bottom of this panel.

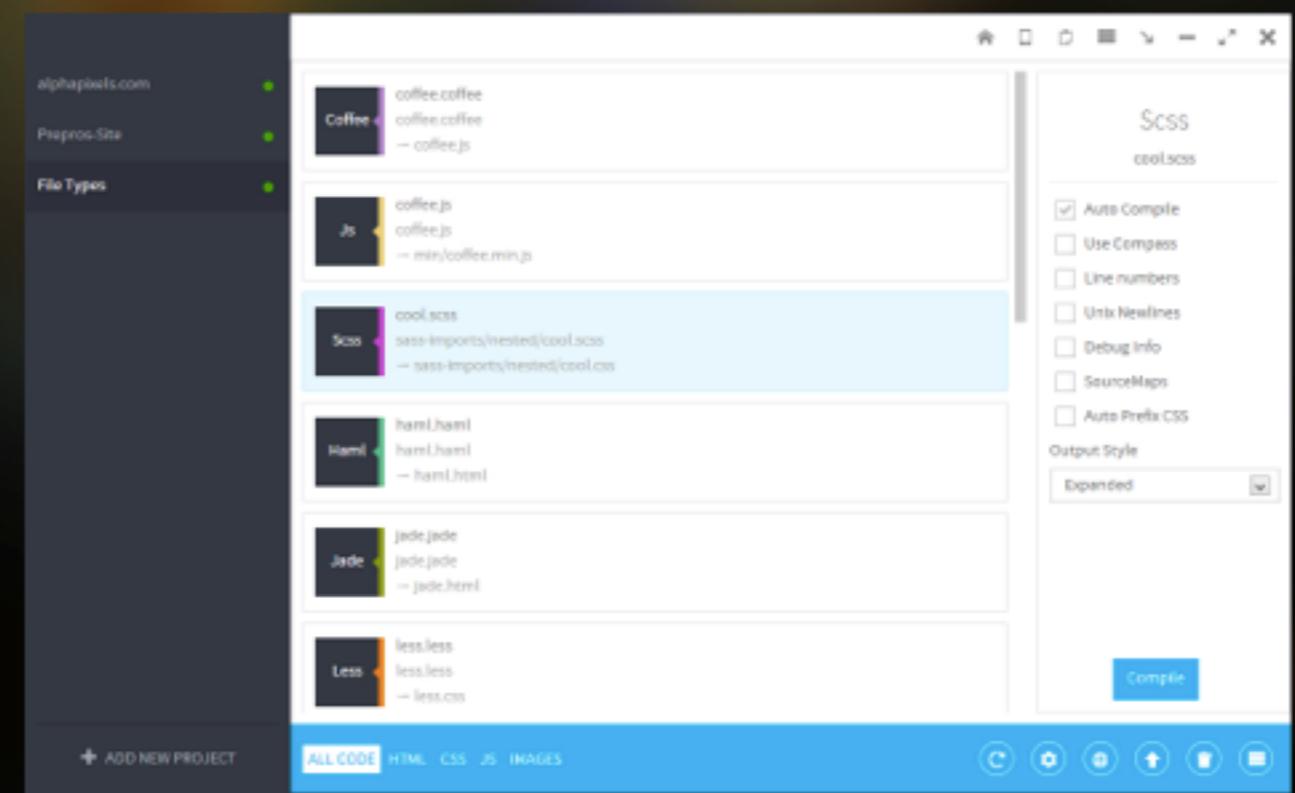
It's like steroids for web developers.

Prepros App

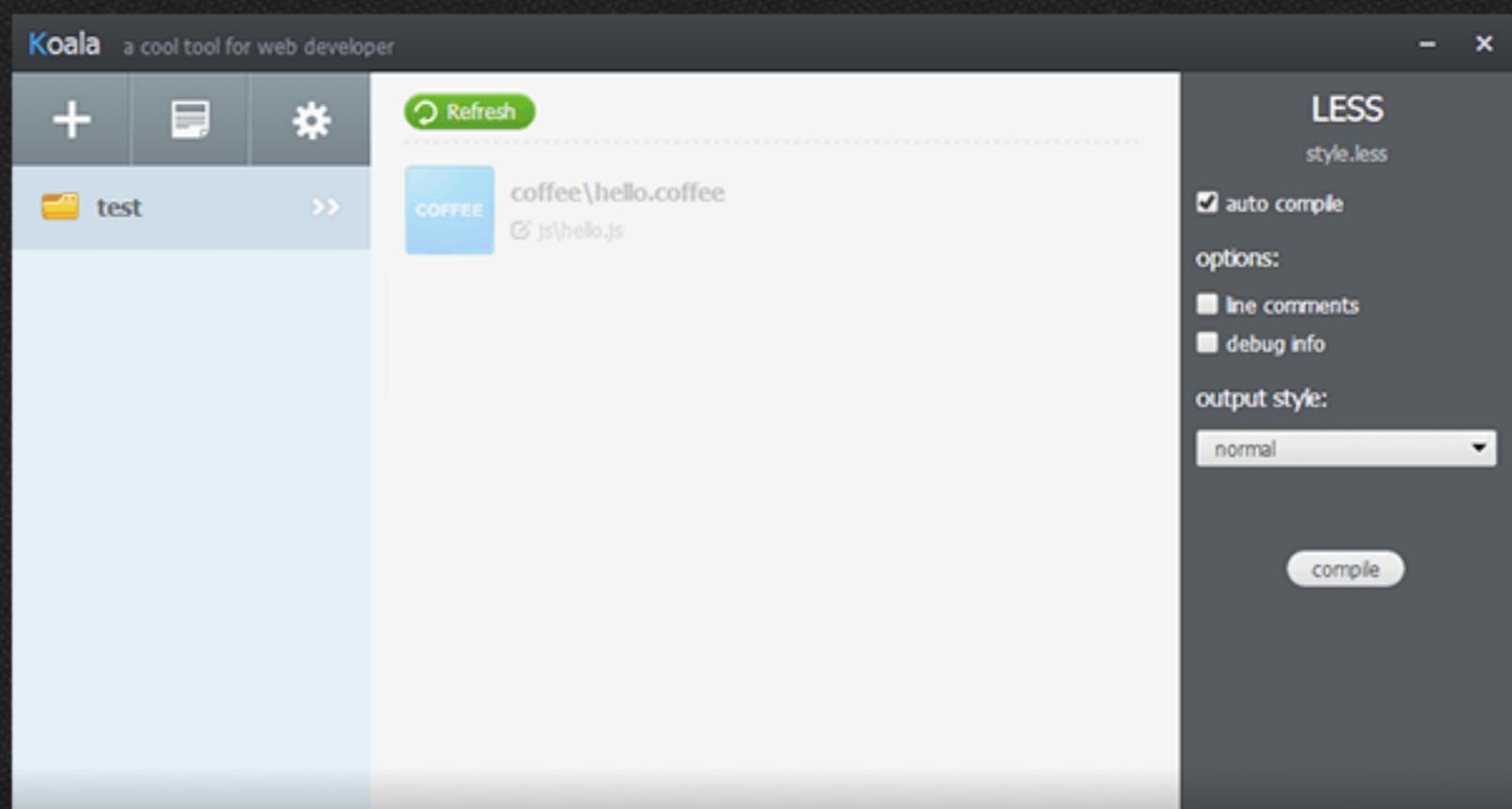
Preprocessing. Live Browser Refreshing. Multi Device Testing and Much More.

Prepros 4.0.1 Windows XP+

Prepros 4.0.1 OSX 10.8+



Koala is a GUI application for Less, Sass, Compass and CoffeeScript compilation, to help web developers to use them more efficiently. Koala can run in windows, linux and mac.

[Other system versions](#) [Donate](#) [Star 825](#) [+1 392](#) [Tweet 506](#) [Like 470](#)

Core Features



Multi-language Support

Support for Less,Sass, CoffeeScript and Compass Framework.



Real-time Compilation

Listening files, compile automatically when the file changes, that everything is



Compile Options

Support set the compiler options for each file.

A quick rehash of [Tobias Ahlin's](#) css trick for [animating link underlines](#). This is a pretty smooth hover trick for links. [Jenn Lukas](#) would be pleased.

Original Example

Hover this link

The only thing it's missing is the a little affordance to help a link look a bit more like a link and be accessible via the keyboard.

Accessible Example

HTML (Slim)

```
1 p A quick rehash of <a  
2 href="https://twitter.com/tobiasahlin">T  
3 obias Ahlin's</a> css trick for <a  
4 href="http://tobiasahlin.com/blog/css-  
5 trickAnimatingLink-  
6 underlines/">animating link  
7 underlines</a>. This is a pretty smooth  
8 hover trick for links. <a  
9 href="http://fuckyeahhovers.tumblr.com/">  
Jenn Lukas</a> would be pleased.
```

CSS (SCSS)

```
1 // -----  
2 // The good stuff  
3 // -----  
4 a.happy-hover {  
5   position: relative;  
6   -webkit-backface-visibility: hidden;  
7   // prevents an annoying flicker on  
8   // hover
```

JS

```
1
```

COMMAND LINE

COMMAND LINE

Sass

```
$ gem install sass  
$ mv style.css style.scss  
$ sass style.scss style.css
```

COMMAND LINE

Less

```
<install node>
<install npm>
$ npm install less
$ lessc style.less
```

COMMAND LINE

Stylus

```
<install node>
<install npm>
$ npm install stylus
$ stylus -c style.styl
```



ОННАІ



A woman with blonde hair and a visible wound on her forehead is looking intently at a computer monitor. The monitor displays a dark, abstract scene with blue and white horizontal bands. She is wearing a light-colored jacket over a dark shirt.

**It's a UNIX system.
I know this.**





The designer's guide to the OSX command prompt

A tutorial for the modern web designer

The command prompt. Once the lofty domain of *that guy you know* with the computer science degree. Now more and more the every day domain of the *hacker web designer*.

Perhaps you've mastered a little Javascript or PHP, but you are realizing that the cool kids are playing around with stuff that is only accessible to people who are comfortable with the command prompt. Or, perhaps you are just curious.

COMPILING

COMPILING

Local

COMPILING

On Deploy

COMPILING

On Request

COMPILING

*On the Client
(don't do this)*

SYNTAX

SYNTAX

Sass

```
.widget {  
  margin: 20px 10px;  
}
```

```
.widget  
  margin: 20px 10px
```

SYNTAX

Less

```
.widget {  
    margin: 20px 10px;  
}
```

SYNTAX

Stylus

```
.widget {  
  margin: 20px 10px;  
}
```

```
.widget  
  margin 20px 10px
```

BEGINNER

BEGINNER

Creating a pre-processed file

Just change the file extension

style.css > style.scss | style.less | style.styl

```
$colorPrimary: #333;  
$siteWidth: 960px;
```

```
body {  
  color: $colorPrimary;  
  width: $siteWidth;  
}
```

```
@colorPrimary: #333;  
@siteWidth: 960px;  
  
body {  
  color: @colorPrimary;  
  width: @siteWidth;  
}
```

BEGINNER

Variables – Stylus

```
colorPrimary #333  
siteWidth 960px
```

```
body  
  color colorPrimary  
  width siteWidth
```

BEGINNER

Variables – Output

```
body {  
    color: #333;  
    width: 960px;  
}
```

BEGINNER

Nesting - CSS

```
nav {  
    width: 200px;  
}  
nav ul {  
    list-style: none;  
}  
nav ul li {  
    background: #ccc;  
}  
nav ul li a{  
    color: #333;  
}
```

BEGINNER

Nesting - Sass/Less

```
nav {  
    width: 200px;  
  
    ul {  
        list-style: none;  
  
        li {  
            background: #ccc;  
  
            a {  
                color: #ccc;  
            }  
        }  
    }  
}
```

BEGINNER

Nesting - Stylus

```
nav
  width 200px

  ul
    list-style none

    li
      background #ccc

      a
        color #ccc
```

BEGINNER

Nesting - Output

```
nav {  
    width: 200px;  
}  
nav ul {  
    list-style: none;  
}  
nav ul li {  
    background: #ccc;  
}  
nav ul li a{  
    color: #333;  
}
```

Pay attention to specificity

BEGINNER

Nesting - Sass/Less

```
nav {  
  width: 200px;  
  
  ul {  
    list-style: none;  
  }  
  li {  
    background: #ccc;  
  }  
  a {  
    color: #ccc;  
  }  
}
```

BEGINNER

Nesting - Stylus

```
nav  
  width 200px
```

```
ul  
  list-style none
```

```
li  
  background #ccc
```

```
a  
  color #ccc
```

BEGINNER

Nesting - Output

```
nav {  
    width: 200px;  
}  
nav ul {  
    list-style: none;  
}  
nav li {  
    background: #ccc;  
}  
nav a{  
    color: #333;  
}
```

BEGINNER

Reference Selector

```
nav {  
    margin: 0; padding: 20px;  
  
    a{  
        color: #000;  
    }  
    a:hover,  
    a:focus {  
        color: #999;  
    }  
    a:active {  
        color: #333;  
    }  
}
```

BEGINNER

Reference Selector - Sass/Less

```
nav {  
  margin: 0;  
  padding: 20px;  
  
  a{  
    color: #000;  
    &:hover,  
    &:focus {  
      color: #999;  
    }  
  
    &:active {  
      color: #333;  
    }  
  }  
}
```

BEGINNER

Reference Selector - Stylus

```
nav
  margin 0
  padding 20px

a
  color #000

  &:hover,
  &:focus
    color #999

  &:active
    color #333
```

BEGINNER

Reference Selector - Output

```
nav {  
    margin: 0;  
    padding: 20px;  
}  
  
nav a{  
    color: #000;  
}  
  
nav a:hover,  
nav a:focus {  
    color: #999;  
}  
  
nav a:active {  
    color: #333;  
}
```

BEGINNER

Reference Selector - Sass/Less

```
nav {  
  margin: 0;  
  padding: 20px;  
  
  .ie6 & {  
    padding: 10px;  
  }  
  .ie7 & {  
    padding: 20px;  
  }  
  .touch & {  
    width: 100%;  
  }  
}
```

BEGINNER

Reference Selector - Stylus

```
nav
```

```
  margin 0  
  padding 20px
```

```
.ie6 &
```

```
  padding 10px
```

```
.ie7 &
```

```
  padding 20px
```

```
.touch &
```

```
  width 100%
```

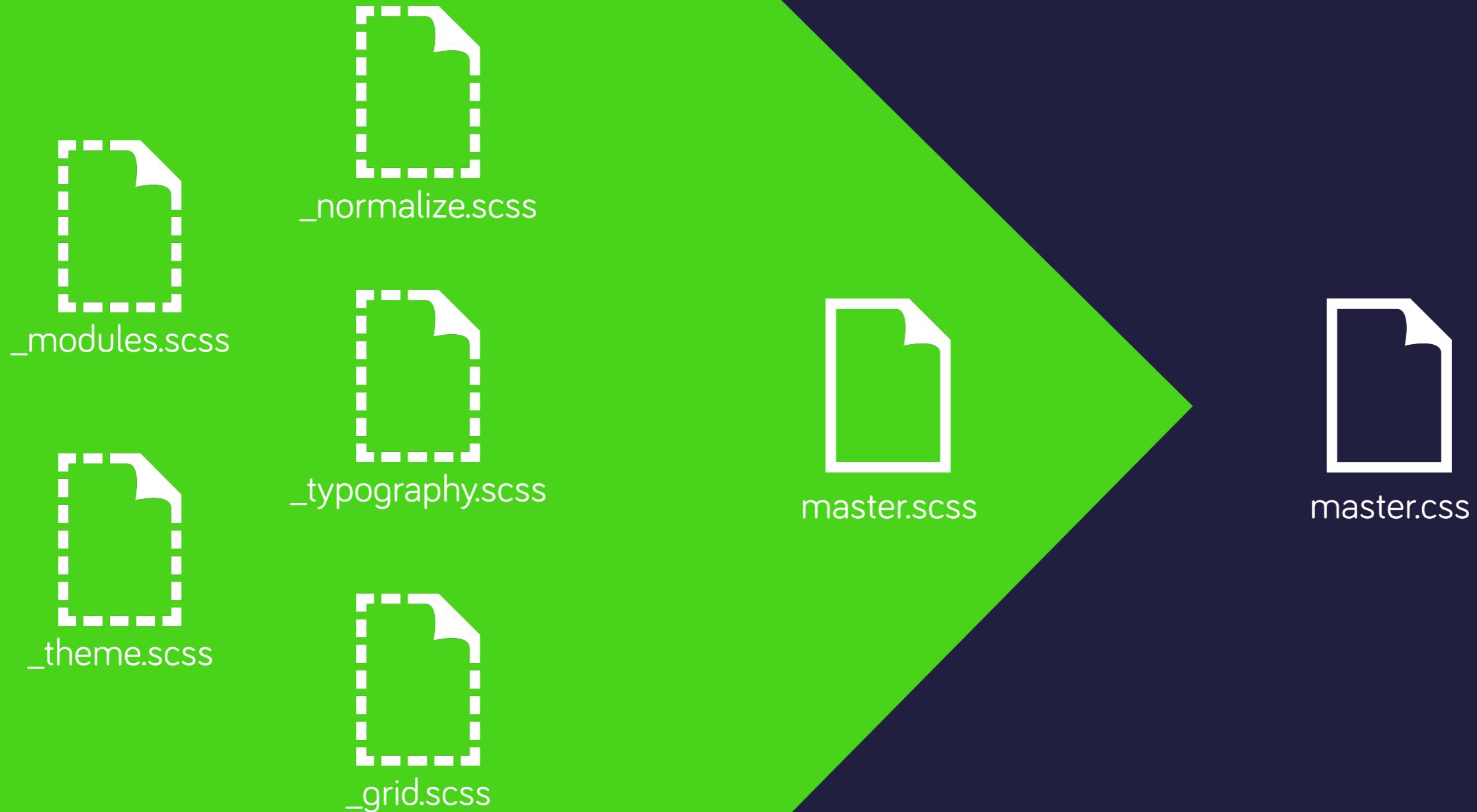
BEGINNER

Reference Selector - Output

```
nav {  
  margin: 0;  
  padding: 20px;  
}  
  
.ie6 nav{  
  padding: 10px;  
}  
  
.ie7 nav {  
  padding: 20px;  
}  
  
.touch nav {  
  width: 100%;  
}
```

BEGINNER

@import



BEGINNER

@import – Sass/Less

```
@import "setup";  
@import "reset";  
@import "base";  
@import "layout";  
@import "typography";  
@import "theme";
```

BEGINNER

@import – Stylus

```
@import "setup"  
@import "reset"  
@import "base"  
@import "layout"  
@import "typography"  
@import "theme"
```

INTERMEDIATE

INTERMEDIATE

Mixins - CSS

```
.box {  
    -webkit-border-radius: 5px;  
    -moz-border-radius: 5px;  
    -ms-border-radius: 5px;  
    -o-border-radius: 5px;  
    border-radius: 5px;  
}
```

INTERMEDIATE

Mixins - Sass

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  -o-border-radius: $radius;  
  border-radius: $radius;  
}  
  
.box {  
  @include border-radius(5px);  
}
```

INTERMEDIATE

Mixins - Less

```
.border-radius(@radius) {  
  -webkit-border-radius: @radius;  
  -moz-border-radius: @radius;  
  -ms-border-radius: @radius;  
  -o-border-radius: @radius;  
  border-radius: @radius;  
}  
  
.box {  
  .border-radius(5px);  
}
```

```
border-radius radius  
-webkit-border-radius radius  
-moz-border-radius radius  
-ms-border-radius radius  
-o-border-radius radius  
border-radius radius
```

```
.box  
border-radius 5px
```

INTERMEDIATE

Mixins - Output

```
.box {  
    -webkit-border-radius: 5px;  
    -moz-border-radius: 5px;  
    -ms-border-radius: 5px;  
    -o-border-radius: 5px;  
    border-radius: 5px;  
}
```

INTERMEDIATE

@extend – Sass

```
.borders {  
  border: 1px solid #efefef;  
  padding: 10px;  
}  
  
p {  
  @extend .borders;  
  font-size: 20px;  
}  
  
ul, ol {  
  @extend .borders;  
  text-transform: uppercase;  
}
```

INTERMEDIATE

:extend – Less

```
.borders {  
    border: 1px solid #efefef;  
    padding: 10px;  
}  
  
p {  
    &:extend(.borders);  
    font-size: 20px;  
}  
  
ul, ol {  
    &:extend(.borders);  
    text-transform: uppercase;  
}
```

INTERMEDIATE

@extend – Stylus

```
.borders  
border 1px solid #efefef  
padding 10px
```

```
p  
@extend .borders  
font-size 20px
```

```
ul, ol  
@extend .borders  
text-transform uppercase
```

INTERMEDIATE

@extend - Output

```
.borders, p, ul, ol {  
  border: 1px solid #efefef;  
  padding: 10px;  
}  
  
p {  
  font-size: 20px;  
}  
  
ul, ol {  
  text-transform: uppercase;  
}
```

INTERMEDIATE

@extend + Placeholder – Sass

```
%borders {  
  border: 1px solid #efefef;  
  padding: 10px;  
}  
  
p {  
  @extend %borders;  
  font-size: 20px;  
}  
  
ul, ol {  
  @extend %borders;  
  text-transform: uppercase;  
}
```

INTERMEDIATE

@extend + Placeholder – Stylus

```
%borders  
border 1px solid #efefef  
padding 10px
```

```
p  
@extend %borders  
font-size 20px
```

```
ul, ol  
@extend %borders  
text-transform uppercase
```

INTERMEDIATE

@extend – Stylus

```
p, ul, ol {  
    border: 1px solid #efefef;  
    padding: 10px;  
}  
  
p {  
    font-size: 20px;  
}  
  
ul, ol {  
    text-transform: uppercase;  
}
```

ADVANCED

ADVANCED

Functions – Sass

```
$grid-columns: 12;
$grid-width: 960px;

@function calculate-column-width($cols) {
  @return ((grid-width / grid-columns) * $cols / grid-width) * 100%;
}

#container {
  margin: 0 auto;
  width: 100%;
}

article {
  float: left;
  width: calculate-column-width(8);
}

aside {
  float: right;
  width: calculate-column-width(4);
}
```

ADVANCED

Functions – Less

```
@grid-columns: 12;
@grid-width: 960px;

.calculate-column-width(@cols) {
  width: (((@grid-width / @grid-columns) * @cols / @grid-width) * 100%);
}

#container {
  margin: 0 auto;
  width: 100%;
}

article {
  float: left;
  .calculate-column-width(8);
}

aside {
  float: right;
  .calculate-column-width(4);
}
```

ADVANCED

Functions – Stylus

```
grid-columns 12  
grid-width 960px
```

```
calculate-column-width(cols)  
((grid-width / grid-columns) * cols / grid-width) * 100%
```

```
#container  
margin 0 auto  
width 100%
```

```
article  
float left  
width calculate-column-width(8)
```

```
aside  
float right  
width calculate-column-width(4)
```

ADVANCED

Functions – Output

```
#container {  
    margin: 0 auto;  
    width: 100%;  
}  
  
article {  
    float: left;  
    width: 66.66667%;  
}  
  
aside {  
    float: right;  
    width: 33.33333%;  
}
```

ADVANCED

Color Functions – Sass

```
rgba($color, $alpha)
hue($color)
saturation($color)
lightness($color)
adjust-hue($color, $degrees)
lighten($color, $amount)
darken($color, $amount)
saturate($color, $amount)
desaturate($color, $amount)
grayscale($color)
complement($color)
invert($color)
```

ADVANCED

Color Functions – Less

```
lighten(@color, amount);
darken(@color, amount);
saturate(@color, amount);
desaturate(@color, amount);
fadein(@color, amount);
fadeout(@color, amount);
fade(@color, amount);
spin(@color, amount);
```

ADVANCED

Color Functions – Stylus

```
rgba(color, alpha)
lighten(color, amount)
darken(color, amount)
desaturate(color, amount)
saturation(color, amount)
invert(color)
hue(color)
saturation(color)
lightness(color)
```

ADVANCED

Control Directives – Sass

```
@if  
@else if  
@else  
@then  
@for  
@each  
@while
```

ADVANCED

Control Directives – Less

when



ADVANCED

Control Directives – Stylus

if
else if
else
unless
for

ADVANCED

if, else – Sass

```
@mixin buttons($color, $type) {  
  @if $type == "flat" {  
    background-color: $color;  
  } @else if $type == "gradient" {  
    background: linear-gradient($color, darken($color, 20%));  
  } @else if $type == "glossy" {  
    background: linear-gradient($color 50%,darken($color, 20%) 50%);  
  } @else {  
    background-color: $color;  
  }  
}  
.button {  
  @include buttons(green, glossy);  
}
```

ADVANCED

when (if, else) – Less

```
.buttons (@color, @type) when (@type == "flat") {  
  background-color: @color;  
}  
.buttons (@color, @type) when (@type == "gradient") {  
  background: linear-gradient(@color, darken(@color, 20%));  
}  
.buttons (@color, @type) when (@type == "glossy") {  
  background: linear-gradient(@color 50%, darken(@color, 20%) 50%);  
}  
  
.button {  
  .buttons(green, glossy);  
}
```

ADVANCED

if, else – Stylus

```
buttons(color, type)
  if type == "flat"
    background-color color
  else if type == "gradient"
    background linear-gradient(color, darken(color, 20%))
  else if type == "glossy"
    background linear-gradient(color 50%,darken(color, 20%) 50%)
  else
    background-color color

.button
  buttons(green, glossy)
```

ADVANCED

if, else – Output

```
.button {  
  background: linear-gradient(#008000 50%, #001a00 50%);  
}
```

ADVANCED

for loop - Sass

```
@for $i from 1px to 5px {  
  .border-#{$i} {  
    border: $i solid #000;  
  }  
}
```

ADVANCED

for loop - Output

```
.border-1px {  
    border: 1px solid black;  
}  
  
.border-2px {  
    border: 2px solid black;  
}  
  
.border-3px {  
    border: 3px solid black;  
}  
  
.border-4px {  
    border: 4px solid black;  
}
```

ADVANCED

each loop - Sass

```
$emotions: happy sad excited mustached;  
  
@each $emotion in $emotions {  
  .feeling-#{$emotion} {  
    background-image: url("images/feeling-#{$emotion}");  
  }  
}
```

ADVANCED

each loop - Output

```
.feeling-happy {  
    background-image: url("images/feeling-happy");  
}  
  
.feeling-sad {  
    background-image: url("images/feeling-sad");  
}  
  
.feeling-excited {  
    background-image: url("images/feeling-excited");  
}  
  
.feeling-mustached {  
    background-image: url("images/feeling-mustached");  
}
```

ADVANCED

Media Queries – Sass

```
$small-breakpoint: 480px;  
$medium-breakpoint: 768px;  
$large-breakpoint: 1024px;  
  
aside {  
    width: 100%;  
    @media screen and (min-width: $small-breakpoint) {  
        width: 100px;  
        float: right;  
    }  
    @media screen and (min-width: $medium-breakpoint) {  
        width: 200px;  
    }  
    @media screen and (min-width: $large-breakpoint) {  
        width: 400px;  
    }  
}
```

ADVANCED

Media Queries – Less

```
@small-breakpoint: 480px;  
@medium-breakpoint: 768px;  
@large-breakpoint: 1024px;  
  
aside {  
    width: 100%;  
    @media screen and (min-width: @small-breakpoint) {  
        width: 100px;  
        float: right;  
    }  
    @media screen and (min-width: @medium-breakpoint) {  
        width: 200px;  
    }  
    @media screen and (min-width: @large-breakpoint) {  
        width: 400px;  
    }  
}
```

ADVANCED

Media Queries – Stylus

```
small-breakpoint 480px  
medium-breakpoint 768px  
large-breakpoint 1024px
```

```
aside  
width 100%
```

```
@media screen and (min-width small-breakpoint)  
width 100px  
float right
```

```
@media screen and (min-width medium-breakpoint)  
width 200px
```

```
@media screen and (min-width large-breakpoint)  
width 400px
```

ADVANCED

Media Queries – Output

```
aside {  
    width: 100%;  
}  
@media screen and (min-width: 480px) {  
    aside {  
        width: 100px;  
        float: right;  
    }  
}  
@media screen and (min-width: 768px) {  
    aside {  
        width: 200px;  
    }  
}  
@media screen and (min-width: 1024px) {  
    aside {  
        width: 400px;  
    }  
}
```

ADVANCED

@content - Sass

```
@mixin respond-to($name){  
  @if $name == small-screen {  
    @media only screen and (min-width: 320px) {  
      @content  
    }  
  }  
  
  @if $name == large-screen {  
    @media only screen and (min-width: 960px) {  
      @content  
    }  
  }  
}  
  
aside {  
  width: 25%;  
  @include respond-to(small-screen) {  
    width: 100%;  
  }  
}
```

ADVANCED

@content – Output

```
aside {  
    width: 25%  
}  
  
@media only screen and (min-width: 320px) {  
    aside {  
        width: 100%  
    }  
}
```

ADVANCED

Hash Maps - Sass

```
$icons: (  
    home: e601,  
    about: e602,  
    services: e603  
);  
  
@each $icon-name, $icon-keycode in $icons {  
    .icon-#$icon-name {  
        &:before {  
            content: "#{$icon-keycode}";  
        }  
    }  
}
```

ADVANCED

Hash Maps – Stylus

```
icons = {  
  home: e601,  
  about: e602,  
  services: e603  
}
```

```
for icon-name, icon-keycode in icons {  
  .icon-{icon-name} {  
    &:before {  
      content: "\{icon-keycode\}";  
    }  
  }  
}
```

ADVANCED

Hash Maps – Output

```
.icon-home:before {  
  content: "\e601";  
}  
  
.icon-about:before {  
  content: "\e602";  
}  
  
.icon-services:before {  
  content: "\e603";  
}
```

FRAMEWORKS



Compass is an open-source *CSS Authoring Framework.*

❤ Why designers love Compass.

1. Experience cleaner markup without presentational classes.
2. It's chock full of the web's best [reusable patterns](#).
3. It makes creating [sprites](#) a breeze.
4. Compass mixins make [CSS3](#) easy.
5. Create beautiful [typographic rhythms](#).
6. Download and create extensions with ease.

✓ Compass uses Sass.

[Sass](#) is an extension of CSS3 which adds nested rules, variables, mixins, selector inheritance, and more. Sass generates well formatted CSS and makes your stylesheets easier to organize and maintain.

Power tools for the web

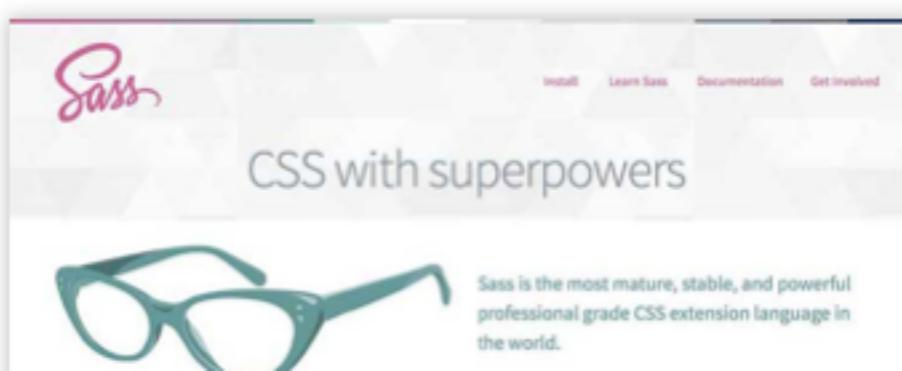
Stable Release: **1.0.9** [Release Notes](#) [Fork on GitHub](#)

Release Candidate: **2.0.0.rc.1** [Release Notes](#)

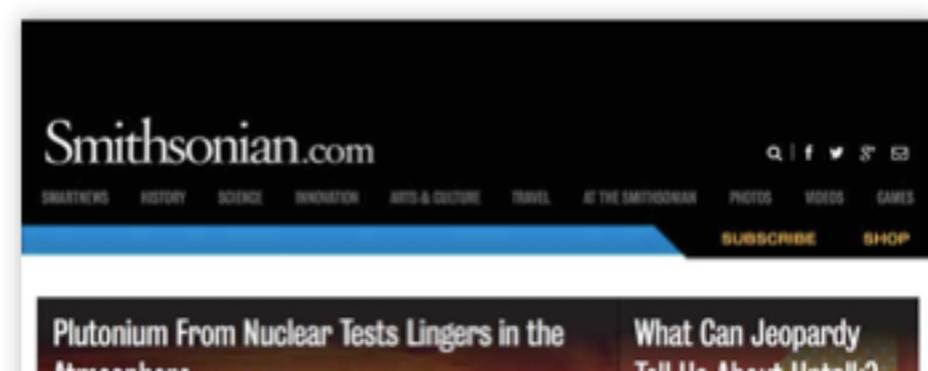
YOUR MARKUP, YOUR DESIGN, YOUR OPINIONS | OUR MATH.

In a world of agile development and super-tablet-multi-magic-laptop-phones, the best layouts can't be contained in a single framework or technique. CSS Libraries are a bloated mess of opinions about how to do your job. Why let the table-saw tell you where to put the kitchen?

IN THE WILD



The screenshot shows the official Sass website. At the top left is the "Sass" logo. To its right are four small navigation links: "Install", "Learn Sass", "Documentation", and "Get Involved". Below the logo, the text "CSS with superpowers" is displayed. A large, stylized illustration of a pair of teal-rimmed glasses occupies the lower-left portion of the page. In the bottom right corner of the page content area, there is a small block of text: "Sass is the most mature, stable, and powerful professional grade CSS extension language in the world."



The screenshot shows the Smithsonian.com homepage. The top half of the page has a dark background with the "Smithsonian.com" logo in white. Below this is a horizontal navigation bar with categories: SMARTNEWS, HISTORY, SCIENCE, INNOVATION, ARTS & CULTURE, TRAVEL, AT THE SMITHSONIAN, PHOTOS, VIDEOS, and GAMES. To the right of the navigation bar are icons for search, social media (Facebook, Twitter, etc.), and a "SUBSCRIBE" button. The main content area below the navigation bar features two news articles: "Plutonium From Nuclear Tests Lingers in the Atmosphere" and "What Can Jeopardy Tell Us About Watson?".



A simple and lightweight mixin library for Sass.

```
ol {  
  > li {  
    background-color: #F4F5F7;  
    @include background-image( linear-gradient(to left, #FFF, #F4F5F7) );  
    float: left;  
    font-family: $lucida-grande;  
    position: relative;  
    @include transition( all 0.8s );  
  
    &::after {  
      content: " ";  
      height: 23px;  
      @include position( absolute, 10px -13px 0 0 );  
      @include transform( rotate(45deg) skew(20deg, 30deg) );  
      width: 23px;  
    }  
  }  
}
```

A KICK-ASS LESS MIXIN LIBRARY FOR EVERYONE.

Support us with [Tweet](#) 572 and [Star](#) 1,187 or [Flattr](#)



LESS Hat 2. Get MORE of LESS with 86 smart mixins.

[▼ DOWNLOAD LESS HAT 2.0.15](#)

[View documentation](#)



Bourbon

NEAT

EXAMPLES

DOCS

Fork me on GitHub

A **lightweight** semantic **grid** framework for **Sass** and **Bourbon**.

<!-- HTML markup for the section right below this code block -->

```
<section>
```

```
  <aside>What is it about?</aside>
```

```
  <article>Neat is an open source semantic grid framework built on top of Sass and Bourbon...</article>
```

```
</section>
```

```
// Enter Neat
```

```
section {
```

```
  @include outer-container;
```

nib

CSS3 extensions for Stylus

Nib is a small and powerful library for the [Stylus CSS language](#), providing robust cross-browser CSS3 mixins to make your life as a designer easier.

```
body {  
  background: linear-gradient(top, white, black);  
}
```

```
body {  
  background: -webkit-gradient(linear,  
    left top,  
    left bottom,  
    color-stop(0, #fff),  
    color-stop(1, #000));  
  background: -webkit-linear-gradient(top, #fff 0%, #000 100%);  
  background: -moz-linear-gradient(top, #fff 0%, #000 100%);  
  background: linear-gradient(top, #fff 0%, #000 100%);  
}
```

axis ⚡ better css

introduction

[introduction](#)

[settings](#)

[reset](#)

[vendor](#)

[positioning](#)

[grid](#)

[utilities](#)

[typography](#)

[vertical-rhythm](#)

[tables](#)

[gradients](#)

[forms](#)

[code](#)

Roots includes a css library called axis built on top of `stylus`. It contains a lot of functionality and is very large, but it should be an easy transition from regular css, less, or sass. With `stylus`, you can use brackets and semicolons or not, your choice. You can even mix them. Pasting straight css will compile perfectly, and nothing in this library overrides anything in regular css in a destructive manner. In addition, this library adds no extra weight to your css footprint unless you do it specifically using mixins or imports. Axis css is an enhancement of regular css, and you can ease into it gradually, unlike some other libraries which require you to learn entirely new syntax.

Since this library is mixin-based, it's worth briefly discussing mixins here. There are two ways to call a mixin in `stylus`, what I call the *mixin form* or the *native form*. The mixin form has parens and comma-separated arguments while the native form uses a colon and space-separated args.

this is the mixin form: `mixin(arg1, arg2)`

and this is the native form: `mixin: arg1 arg2`

When calling mixins in roots, you should use the native form, *unless the mixin is being called at the root level*,

Foundation

The most advanced responsive front-end framework in the world.

[Download Foundation 5](#)

★ 15.9k stargazers  @foundationzurb



Responsive design gets a whole lot faster for users.

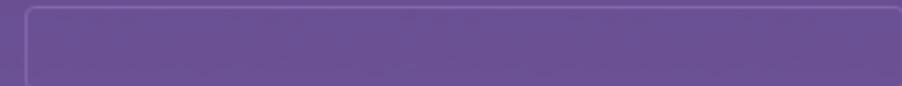
It's now crazy fast for designers and engineers to code and learn too.



The most popular front-end framework for developing responsive, mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.1.1





WHAT?

SASSC

ABOUT

What?

[Sass](#) is a pre-processing language for CSS. It allows you to write cleaner stylesheets and makes collaboration on your CSS a breeze. There's a ton of information on Sass out there, so we won't repeat it all here. Just make sure to check out [the Sass site](#) for tutorials and examples.

Sass was originally written in Ruby. Ruby's great, but people started having a couple of issues. First, we want everyone to enjoy Sass, no matter what language they use. Why restrict everyone to using Ruby? In addition, Ruby can be kind of slow. Lowering compile time for users is important. Enter Libsass.

Libsass is C/C++ port of the Sass engine. The point is to be simple, fast, and easy to integrate. Find out more about the project over at [Github](#).

SassC

fin.