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
What is a Preprocessor?
Expressive, dynamic, robust CSS

CSS needs a hero

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

What if we could omit braces?

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

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

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:

```less
@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) {
  -webkit-box-shadow: @style @c;
  box-shadow: @style @c;
}
```
CSS with superpowers

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

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.
If you write sucky CSS, a pre-processor won’t make it suck less.
QUICK START
It's like steroids for web developers.
Prepros App


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.

Download
Version 2.0.2, 2014-01-24

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.
```bash
$ gem install sass
$ mv style.css style.scss
$ sass style.scss style.css
```
<install node>
<install npm>
$ npm install less
$ lessc style.less
<install node>
<install npm>
$ npm install stylus
$ stylus -c style.style.styl
OH HAI
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 interested in learning...
COMPILING
Local
On Deploy
On Request
On the Client
(don’t do this)
.widget {
  margin: 20px 10px;
}

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

.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;
}

Variables – Sass
@colorPrimary: #333;
@siteWidth: 960px;

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

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

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

    ul {
        list-style: none;

        li {
            background: #ccc;

            a {
                color: #ccc;
            }
        }
    }
}
```
```html
nav
  width 200px

ul
  list-style none

li
  background #ccc

  a
    color #ccc
```
```html
nav {
  width: 200px;
}
nav ul {
  list-style: none;
}
nav ul li {
  background: #ccc;
}
nav ul li a {
  color: #333;
}
```
Pay attention to specificity
```css
nav {
    width: 200px;

    ul {
        list-style: none;
    }

    li {
        background: #ccc;
    }

    a {
        color: #ccc;
    }
}
```
```html
nav
  width 200px

ul
  list-style none

li
  background #ccc

a
  color #ccc
```
```css
nav {
    width: 200px;
}
nav ul {
    list-style: none;
}
nav li {
    background: #ccc;
}
nav a{
    color: #333;
}
```
```css
nav {
    margin: 0; padding: 20px;
    a {
        color: #000;
    }
    a:hover, a:focus {
        color: #999;
    }
    a:active {
        color: #333;
    }
}
```
nav {
  margin: 0;
  padding: 20px;
}

a{
  color: #000;
  &:hover, &:focus {
    color: #999;
  }
  &:active {
    color: #333;
  }
}

Reference Selector - Sass/Less
nav
  margin 0
  padding 20px

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

nav a{
    color: #000;
}

nav a:hover,
nav a:focus {
    color: #999;
}

nav a:active {
    color: #333;
}
nav {
  margin: 0;
  padding: 20px;

  .ie6 & {
    padding: 10px;
  }

  .ie7 & {
    padding: 20px;
  }

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

.ie6 &
  padding 10px

.ie7 &
  padding 20px

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

@import

_modules.scss
 Normalize.scss
 _theme.scss
 Typoigraphy.scss
 Grid.scss
 master.scss
 master.css
@import "setup";
@import "reset";
@import "base";
@import "layout";
@import "typography";
@import "theme";
@import "setup"
@import "reset"
@import "base"
@import "layout"
@import "typography"
@import "theme"
INTERMEDIATE
.box {
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  -ms-border-radius: 5px;
  -o-border-radius: 5px;
  border-radius: 5px;
}
@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);
}
.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

.box
border-radius 5px
.box {
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    -ms-border-radius: 5px;
    -o-border-radius: 5px;
    border-radius: 5px;
}
.borders {
    border: 1px solid #efefef;
    padding: 10px;
}

p {
    @extend .borders;
    font-size: 20px;
}

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

p {
    &:extend(.borders);
    font-size: 20px;
}

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

p
  @extend .borders
  font-size 20px

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

p
  @extend %borders
  font-size 20px

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

p {
    font-size: 20px;
}

ul, ol {
    text-transform: uppercase;
}
ADVANCED
$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);
}
@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);
}
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)
#container {
    margin: 0 auto;
    width: 100%;
}

article {
    float: left;
    width: 66.66667%;
}

aside {
    float: right;
    width: 33.33333%;
}
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)
lighten(@color, amount);
darken(@color, amount);
saturate(@color, amount);
desaturate(@color, amount);
fadein(@color, amount);
fadeout(@color, amount);
fade(@color, amount);
spin(@color, amount);
rgba(color, alpha)
lighen(color, amount)
darken(color, amount)
desaturate(color, amount)
saturate(color, amount)
invert(color)
hue(color)
saturation(color)
lighness(color)
when
if
else if
else
unless
for
@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);
}
.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);
}
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)
.button {
    background: linear-gradient(#008000 50%, #001a00 50%);
}

if, else – Output
@for $i from 1px to 5px {
    .border-#{${$i}} {
        border: $i solid #000;
    }
}
for loop – Output

```css
.border-1px {
  border: 1px solid black;
}
.border-2px {
  border: 2px solid black;
}
.border-3px {
  border: 3px solid black;
}
.border-4px {
  border: 4px solid black;
}
```
$emotions: happy sad excited mustached;

@each $emotion in $emotions {
  .feeling-#$emotion {
    background-image: url("images/feeling-#$emotion");
  }
}
.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");
}
$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;
  }
}
@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;
    }
}
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
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;
  }
}
@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%;
  }
}
aside {
    width: 25%
}

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

$icons: (  
  home: e601,
  about: e602,
  services: e603
);

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

for icon-name, icon-keycode in icons {
    .icon-{icon-name} {
        &:before {
            content: "\{icon-keycode}\";
        }
    }
}
Hash Maps – Output

Advanced

.icon-home:before {
    content: "\e601";
}

.icon-about:before {
    content: "\e602";
}

.icon-services:before {
    content: "\e603";
}
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.

Brilliant people use Compass, including these wildly talented folks:
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

Sass

CSS with superpowers

Smithsonian.com

Plutonium From Nuclear Tests Lingers in the

What Can Jeopardy? TAW: Nineteen Eighty-
A simple and lightweight mixin library for Sass.
A KICK-ASS LESS MIXIN LIBRARY FOR EVERYONE.

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

DOWNLOAD LESS HAT 2.0.15

View documentation
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 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.

```css
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%);
}
```
introduction

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  

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?

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.
fin.