Syntactically Awesome Style Sheets

IRON HACK

Frontend

Sass

CSS preprocessor

IRON HACK

Features

- Nesting selectors
- $variables
- Functions
- Mixins
- @import
- @extend
- @each (loops)
- @if (conditionals)

IRON HACK

gem install sass

IRON HACK

gem install compass

Create a Sass/Compass project

```
$ compass create <your-project>
$ cd <your-project>
```

This command creates a folder with the name of your project and places a set of scss files ready to be used within your HTML.

IRON HACK

Project structure

```
<your-project>
--config.rb
--sass/
  --screen.scss
--stylesheets/
  --screen.css
```

The /sass/ folder contains the .scss files ready to be edited inside. We'll mainly use screen.scss. The .scss files compile to .css in the /stylesheets/ folder.

IRON HACK

Project structure

```
$ mkdir sass/imports
```

Create an imports folder inside /sass/.

IRON HACK

Project configuration

```
require 'compass/import-once/activate'
# Require any additional compass plugins here.

# Set this to the root of your project when deployed:
http_path = "/"
css_dir = "stylesheets"
sass_dir = "sass"
images_dir = "images"
javascripts_dir = "javascripts"
```

Open the config.rb file in your code editor. There's only a couple of active configuration lines requiring a useful library and setting some paths.

IRON HACK

Project configuration

```
# You can select your preferred output style here (can be
overridden via the command line):
output_style = :nested

# To enable relative paths to assets via compass helper
functions. Uncomment:
relative_assets = true
```

Uncomment the `output_style` and the `relative_assets` lines. Set `output_style` to `:nested`. This produces more readable .css files.

IRON HACK

Project configuration

```
# Add the imports path for modules and addons
add_import_path "sass/imports"
```

We'll use the `add_import_path` function to make it easier to import our modules in the future. Place it at the bottom of your config.rb file. Save and we're ready to import some .css and .scss files.

IRON HACK

Exercise

Create a Compass project.

1. Download the Sass version of Bootstrap, place the .scss files in your /sass/ directory, copy the rest of the folders in the root of your project.
2. Create an index.html file in the root of your project.
3. Include screen.css (only this one).
4. Include bootstrap.js.

IRON HACK

Two ways of writing Sass

```
// Strict Sass (file.sass)
// tab based
// no braces or semicolons

.box
  property: value
  property: value
```

```
// SCSS (file.scss)
// CSS compatible
// valid CSS === valid SCSS

.box {
  property: value;
  property: value;
}
```

Comments

Original SCSS

```
// This is a Sass comment

/*
This is a normal
CSS comment
*/

.box {
  property: value;
  property: value;
}
```

Compiled CSS

```
/*
This is a normal
CSS comment
*/

.box {
  property: value;
  property: value;
}
```

@import

Original SCSS

```
// Import SCSS
// sass/imports/_bootstrap.scss
@import "bootstrap"

// Import 3rd party CSS :(
// stylesheets/plugin.css
@import "plugin.css"

html {
  background: #ccc;
}
```

Compiled CSS

```
@import url(plugin.css);
/*
The bootstrap css contents get
embedded here
*/

html {
  background: #ccc;
}
```

@import

Original SCSS

```
// Import SCSS
// sass/imports/_bootstrap.scss
@import "bootstrap"

// Import 3rd party CSS :D
// just rename plugin.css to
_plugin.scss and move it to the
sass/imports/ folder.
// sass/imports/_plugin.scss
@import "plugin"

html {
  background: #ccc;
}
```

Compiled CSS

```
/*
The bootstrap css contents and the
3rd party css get embedded here
*/

html {
  background: #ccc;
}
```

Exercise

@import bootstrap inside your screen.scss file.

Since we're using compass, first @import bootstrap-compass.

IRON HACK

Compile Sass

```
$ cd <your-project>
$ compass compile
```

Run `compass compile` inside your project's root folder.

IRON HACK

Watch for changes and compile

```
$ cd <your-project>
$ compass watch
```

Run `compass watch` inside your project's root folder.
From now on, everytime you save your .scss files
compass will run a `compass compile` automatically.

IRON HACK

Nested selectors

Original SCSS

```
.main-nav {
  padding: 1em;

  > li {
    color: #cc0000;
  }
}
```

Compiled CSS

```
.main-nav {
  padding: 1em;
}

.main-nav > li {
  color: #cc0000;
}
```

Nested selectors (&)

Original SCSS

```scss
.main-nav {
  padding: 1em;

  > li {
    color: #cc0000;
  }

  .footer & {
    padding-top: 0;
  }
}
```

Compiled CSS

```css
.main-nav {
  padding: 1em;
}

.main-nav > li {
  color: #cc0000;
}

.footer .main-nav {
  padding-top: 0;
}
```

$variables

Original SCSS

```scss
$color: #cc0000;

.main-nav {
  padding: 1em;

  > li {
    color: $color;
  }
}

a {
  color: $color;
}
```

Compiled CSS

```css
.main-nav {
  padding: 1em;
}

.main-nav > li {
  color: #cc0000;
}

a {
  color: #cc0000;
}
```

functions

Original SCSS

```scss
// Function definition
@function pxr($num) {
  @return ($num / 16) + rem;
}

// Function usage
.title {
  font-size: pxr(24);
}
```

Compiled CSS

```css
.title {
  font-size: 1.5rem;
}
```

mixins

Original SCSS

```scss
// Mixin definition
@mixin large-text {
  font-family: Arial;
  color: black;
  font-size: 2em;
}

// Mixin usage
.title {
  @include large-text;
}
```

Compiled CSS

```css
.title {
  font-family: Arial;
  font-size: 2em;
  color: black;
}
```

mixins (with arguments)

Original SCSS

```scss
// Mixin definition
@mixin large-text($color, $size) {
  font-family: Arial;
  color: $color;
  font-size: $size;
}

// Mixin usage
.title {
  @include large-text(black, 3em);
}


.subtitle {
  @include large-text(grey, 1.5
em);
}
```

Compiled CSS

```css
.title {
  font-family: Arial;
  color: black;
  font-size: 3em;
}

.title {
  font-family: Arial;
  color: grey;
  font-size: 1.5em;
}
```

mixins (with arguments and defaults)

Original SCSS

```scss
// Mixin definition
@mixin large-text($color, $size:
2em) {
  font-family: Arial;
  color: $color;
  font-size: $size;
}

// Mixin usage
.title {
  @include large-text(black);
}


.subtitle {
  @include large-text(grey, 1em);
}
```

Compiled CSS

```css
.title {
  font-family: Arial;
  color: black;
  font-size: 2em;
}

.title {
  font-family: Arial;
  color: grey;
  font-size: 1em;
}
```

@extend

Original SCSS

```scss
.button-send {
  border: 2px solid blue;
  background: white;
  color: blue;
  font-size: 1.2rem;
}

.button-register {
  @extend .button-send;
  margin-left: 1em;
}
```

Compiled CSS

```css
.button-send,
.button-register {
  border: 2px solid blue;
  background: white;
  color: blue;
  font-size: 1.2rem;
}

.button-register {
  margin-left: 1em;
}
```

@extend (with placeholder selectors)

Original SCSS

```scss
%button-blue {
  border: 2px solid blue;
  background: white;
  color: blue;
  font-size: 1.2rem;
}

.button-register {
  @extend %button-blue;
  margin-left: 1em;
}

.button-cta {
  @extend %button-blue;
  font-size: 1.5em;
}
```

Compiled CSS

```css
.button-register,
.button-cta {
  border: 2px solid blue;
  background: white;
  color: blue;
  font-size: 1.2rem;
}

.button-register {
  margin-left: 1em;
}

.button-cta {
  font-size: 1.5em;
}
```

@each

Original SCSS

```scss
$buttons: blue, red, green, white;

@each $button in $buttons {

  .btn-#{$button} {
    background: $button;
  }

}
```

Compiled CSS

```css
.btn-blue {
  background: blue;
}

.btn-red {
  background: red;
}

.btn-green {
  background: green;
}

.btn-white {
  background: white;
}
```

@if

Original SCSS

```scss
$buttons: blue, red, green, white;

@each $button in $buttons {

  .btn-#{$button} {
    background: $button;

    @if $button == white {
      color: black;
    }
  }
}
```

Compiled CSS

```css
.btn-blue {
  background: blue;
}

.btn-red {
  background: red;
}

.btn-green {
  background: green;
}

.btn-white {
  background: white;
  color: black;
}
```

Exercise