Sass, un preprocesador CSS

Adolfo Sanz De Diego

Septiembre 2016

Table of Contents

# El autor

## Adolfo Sanz De Diego

* Empecé **desarrollando aplicaciones web**, hasta que di el salto a la docencia.
* Actualmente soy **Asesor Técnico Docente** en el servicio TIC de la D.G de Infraestructuras y Servicios de la Consejería de Educación, Juventud y Deporte de la Comunidad de Madrid.
* Además colaboro como **formador especializado en tecnologías de desarrollo**.

## Algunos proyectos

* **Hackathon Lovers** <http://hackathonlovers.com>: un grupo creado para emprendedores y desarrolladores amantes de los hackathones.
* **Password Manager Generator** <http://pasmangen.github.io>: un gestor de contraseñas online.
* **MarkdownSlides** <https://github.com/asanzdiego/markdownslides>: un script para crear slides a partir de ficheros MD.

## ¿Donde encontrarme?

* Mi nick: **asanzdiego**
  + AboutMe: <http://about.me/asanzdiego>
  + GitHub: <http://github.com/asanzdiego>
  + Twitter: <http://twitter.com/asanzdiego>
  + Blog: <http://asanzdiego.blogspot.com.es>
  + LinkedIn: <http://www.linkedin.com/in/asanzdiego>
  + Google+: <http://plus.google.com/+AdolfoSanzDeDiego>

# Introducción

## ¿Qué es?

* Less es un **pre-procesador de CSS**.
* Añade características como **variables, mixins, funciones, etc**.

## Ventajas

* El CSS es así **más fácil de mantener, personalizable y extensible**.
* Sass tiene una **sintaxis parecida a CSS**.
* Es el preprocesador que usa Bootstrap 4.0.

# Instalación

## Instalar Ruby

* Ubuntu:

sudo apt-get install ruby-full

* Windows: <http://rubyinstaller.org/>
* Mac: incluido

## Instalar Sass

* Después de instalar Ruby, ejecutar

sudo gem install sass

## Preprocesar

* Un fichero

sass --watch input.scss:output.css

* Un directorio

sass --watch input/dir:output/dir

## Probar sin instalar

* <http://www.sassmeister.com/>

# Características

## Variables

* El siguiente código:

$font-stack: Helvetica, sans-serif;  
$primary-color: #333;  
  
body {  
 font: 100% $font-stack;  
 color: $primary-color;  
}

* Se compila a:

body {  
 font: 100% Helvetica, sans-serif;  
 color: #333;  
}

## Valores por defecto

* El siguiente código:

$content: "First content";  
$content: "Second content?" !default;  
$new\_content: "First time reference" !default;  
  
#main {  
 content: $content;  
 new-content: $new\_content;  
}

## Valores por defecto

* Se compila a:

#main {  
 content: "First content";  
 new-content: "First time reference";  
}

## Scope

* Los **ámbitos de las variables** en Sass es muy similar a otros lenguajes:

$var: red;  
  
#page {  
 $var: white;  
 #header {  
 color: $var; // white  
 }  
}

## Reglas anidadas

* El siguiente código:

nav {  
 ul {  
 margin: 0;  
 padding: 0;  
 list-style: none;  
 }  
  
 li { display: inline-block; }  
}

* Se compila a:

nav ul {  
 margin: 0;  
 padding: 0;  
 list-style: none;  
}  
  
nav li {  
 display: inline-block;  
}

## Parciales

* Es un archivo con un guión bajo:

\_partial.scss

* Con esto Sass entiende que es un archivo parcial y que no debe generar CSS.

## Imports

* Imaginemos el archivo **\_reset.scss**:

html, body, ul {  
 margin: 0;  
 padding: 0;  
}

* Y el archivo **base.scss**:

@import 'reset';  
  
body {  
 font: 100% Helvetica, sans-serif;  
 background-color: #efefef;  
}

* Se compila a:

html, body, ul, ol {  
 margin: 0;  
 padding: 0;  
}  
  
body {  
 font: 100% Helvetica, sans-serif;  
 background-color: #efefef;  
}

## Parent

* El código '&' hace referencia al padre:

a {  
 font-weight: bold;  
 text-decoration: none;  
 &:hover { text-decoration: underline; }  
 body.firefox & { font-weight: normal; }  
}

* Se compila a:

a {  
 font-weight: bold;  
 text-decoration: none;  
}  
a:hover { text-decoration: underline; }  
body.firefox a { font-weight: normal; }

## Mixins

* El siguiente código:

@mixin border-radius($radius) {  
 -webkit-border-radius: $radius;  
 -moz-border-radius: $radius;  
 -ms-border-radius: $radius;  
 border-radius: $radius;  
}  
  
.box {  
 @include border-radius(10px);  
}

* Se compila a:

.box {  
 -webkit-border-radius: 10px;  
 -moz-border-radius: 10px;  
 -ms-border-radius: 10px;  
 border-radius: 10px;  
}

## Argumentos variables

* El siguiente código:

@mixin box-shadow($shadows...) {  
 -moz-box-shadow: $shadows;  
 -webkit-box-shadow: $shadows;  
 box-shadow: $shadows;  
}  
  
.shadows {  
 @include box-shadow(0px 4px 5px #666, 2px 6px 10px #999);  
}

* Se compila a:

.shadows {  
 -moz-box-shadow: 0px 4px 5px #666, 2px 6px 10px #999;  
 -webkit-box-shadow: 0px 4px 5px #666, 2px 6px 10px #999;  
 box-shadow: 0px 4px 5px #666, 2px 6px 10px #999;  
}

## Pasar contenido a mixin

* El siguiente código:

@mixin apply-to-ie6-only {  
 \* html {  
 @content;  
 }  
}  
@include apply-to-ie6-only {  
 #logo {  
 background-image: url(/logo.gif);  
 }  
}

* Se compila a:

\* html #logo {  
 background-image: url(/logo.gif);  
}

## Interpolación

* Podemos sustituir el palor de una variable en nombre de propiedades o selectores con #{$variable}:

$position: left;  
@mixin car($brand, $color) {  
 .coche.#{$brand} {  
 border-#{$position}: 2px;  
 background-color: $color;  
 background-image: url('img/#{$brand}-#{$color}.png');  
 }  
}  
@include car('audi', 'green');

* Se compila a:

.coche.audi {  
 border-left: 2px;  
 background-color: green;  
 background-image: url('img/audi-green.png');  
}

## Extend

* El siguiente código:

.message {  
 color: #333;  
}  
.success {  
 @extend .message;  
 border-color: green;  
}  
.error {  
 @extend .message;  
 border-color: red;  
}

* Se compila a:

.message, .success, .error, .warning {  
 color: #333;  
}  
.success {  
 border-color: green;  
}  
.error {  
 border-color: red;  
}

## Operadores

* El siguiente código:

article[role="main"] {  
 float: left;  
 width: 600px / 960px \* 100%;  
}  
  
aside[role="complementary"] {  
 float: right;  
 width: 300px / 960px \* 100%;  
}

* Se compila a:

article[role="main"] {  
 float: left;  
 width: 62.5%;  
}  
  
aside[role="complementary"] {  
 float: right;  
 width: 31.25%;  
}

## Funciones

* Sass dispone de una variedad de **funciones matemáticas, que manipulan cadenas, y que transforman los colores**:

$base: #f04615;  
$list: 200, 500, 1200;  
  
.class {  
 width: nth($list, 3);  
 color: darken($base, 5%);  
 background-color:  
 lighten($base, 25%);  
}

* Se compila a:

.class {  
 width: 1200;  
 color: #f6430f;  
 background-color: #f8a58d;  
}

## Definir funciones

* Podemos definir nuestras propias funciones:

$grid-width: 40px;  
$gutter-width: 10px;  
  
@function grid-width($n) {  
 @return $n \* $grid-width + ($n - 1) \* $gutter-width;  
}  
  
#sidebar { width: grid-width(5); }

* Se compila a:

#sidebar {  
 width: 240px;  
}

## IF

* El siguiente código:

$animal: cat;  
p {  
 @if 1+1 == 2 { border: 2px solid black; }  
 @if $animal == cat { color: white; }  
}

## IF

* Se compila a:

p {  
 border: 2px solid black;  
 color: white;  
}

## FOR

* El siguiente código:

@for $i from 1 to 4 {  
 .item-#{$i} { width: 2em \* $i; }  
}

* Se compila a:

.item-1 { width: 2em; }  
.item-2 { width: 4em; }  
.item-3 { width: 6em; }

## EACH

* El siguiente código:

@each $animal in puma, tiger, salamander {  
 .#{$animal}-icon {  
 background-image: url('/images/#{$animal}.png');  
 }  
}

* Se compila a:

.puma-icon {  
 background-image: url('/images/puma.png'); }  
.tiger-icon {  
 background-image: url('/images/tiger.png'); }  
.salamander-icon {  
 background-image: url('/images/salamander.png'); }

## WHILE

* El siguiente código:

$i: 6;  
@while $i > 0 {  
 .item-#{$i} { width: 2em \* $i; }  
 $i: $i - 2;  
}

* Se compila a:

.item-6 { width: 12em; }  
.item-4 { width: 8em; }  
.item-2 { width: 4em; }

## logs

* El siguiente código:

@debug 10em + 12em;

* Saca por pantalla a la hora de compilar:

Line 1 DEBUG: 22em

# Acerca de

## Licencia

* Estas **transparencias** están hechas con:
  + MarkdownSlides: <https://github.com/asanzdiego/markdownslides>
* Estas **transparencias** están bajo una licencia Creative Commons Reconocimiento-CompartirIgual 3.0:
  + <http://creativecommons.org/licenses/by-sa/3.0/es>

## Fuentes

* Transparencias:
  + <https://github.com/asanzdiego/curso-interfaces-web-2016/tree/master/04-preprocesadores-css/04.02-sass/slides>
* Ejercicios:
  + <https://github.com/asanzdiego/curso-interfaces-web-2016/tree/master/04-preprocesadores-css/04.02-sass/src>

## Bibliografía

* Documentación oficial de Sass
  + <http://sass-lang.com/>
* Para probar Sass
  + <http://www.sassmeister.com/>