

SASSIONS



Escribe menos CSS con Sass

<http://sass-lang.com/>

¿POR QUÉ?

CSS es nuestra herramienta de batalla para ordenar y dar una capa visual a la web; pero se puede convertir en un auténtico dolor de cabeza cuando los archivos crecen mucho.

El problema más común al trabajar con css son las **contínuas repeticiones de código** (colores, selectores anidados, vendors,...)

```
.container{
    border-color : #fabada;
    color : #fabada;
    -webkit-box-sizing : border-box;
    -moz-box-sizing : border-box;
    -o-box-sizing : border-box;
    -ms-box-sizing : border-box;
    box-sizing : border-box;
}
.container p {
    -webkit-box-sizing : border-box;
    -moz-box-sizing : border-box;
    -o-box-sizing : border-box;
    -ms-box-sizing : border-box;
    box-sizing : border-box;
}
```

¿POR QUÉ?



El mismo código

```
$mainColor: #fabada;
@mixin boxSizing($val:border-box){
  -webkit-box-sizing: $val;
  -moz-box-sizing: $val;
  -o-box-sizing: $val;
  -ms-box-sizing: $val;
  box-sizing: $val;
}

.container{
  border-color: $mainColor;
  color: $mainColor;
  @include boxSizing();
  p {
    @include boxSizing();
  }
}
```

ALTERNATIVAS

- LESS
- Stylus

INSTALACIÓN

Sass está desarrollado en Ruby, y generado como una gema, lo que permite su instalación mediante el administrador de paquetes (package manager) de ruby `gem`, por tanto, para poder trabajar con sass es necesario tener instalado Ruby.

INSTALANDO EN



Ruby Installer (<http://www.rubyinstaller.org/>)

facilita la instalación de Ruby mediante un cómodo instalador

Una vez acabada la instalación de ruby instalamos sass

```
$ gem install sass
```

Verificamos que se encuentra instalado correctamente

```
$ sass -v
```

INSTALANDO EN

Ruby viene instalado por defecto en OSX, sólo hay que instalar la gema de Sass

```
$ gem install sass
```

Verificamos que se encuentra instalado correctamente

```
$ sass -v
```

PRIMEROS PASOS

Sass soporta dos formatos de escritura **.sass**(versión antigua) y **.scss** (nos centraremos en ésta)

- Procesar un archivo

```
$ sass input.scss output.css
```

- Escuchar los cambios de un archivo y procesarlo

```
$ sass --watch input.scss:output.css
```

- Escuchar un directorio entero y procesarlo en otro directorio

```
$ sass --watch app/sass:public/stylesheets
```

- Convertir .css a .scss

```
$ sass-convert style.css style.scss --from css --to scss
```


CONFIGURANDO LA SALIDA

```
$ sass input.scss output.css --style [nested|expanded|compact|compressed]
```

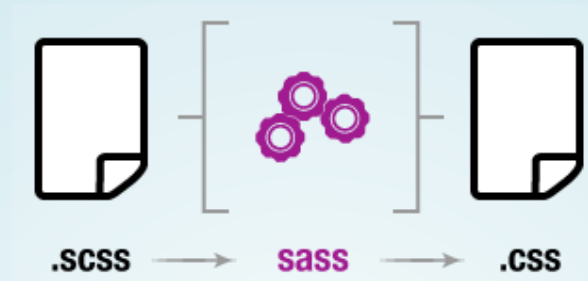
- **nested** (por defecto)
Cada propiedad tiene su propia línea. Cada regla es indentada en base de su profundidad
- **expanded**
La más común. Cada regla y propiedad tienen su propia línea.
- **compact**
Cada regla tiene su propia línea, las propiedades se apilan una detrás de la otra en una sola línea.
Las reglas enlazadas van en líneas consecutivas.
- **compressed**
Ésta opción toma la cantidad mínima de espacio posible, eliminando los espacios innecesarios (los espacios entre selectores son necesarios) y retornos de carro.

CARÁCTERÍSTICAS DE SASS

Sass son los mismos css de siempre pero con superpoderes

- Preprocesamiento (.scss --> .css)
- Variables
- Nesting (Selectores enlazados)
- Import (importado de archivos)
- Partials (archivos parciales)
- Mixins (funciones)
- Herencia (Herencia de clases)
- Operaciones (+, -, /, * ...)
- Directivas de control (@if, @for, ...)

PREPROCESAMIENTO



Sass lee los archivos con extensión **.scss** escritos en su meta-lenguaje y los procesa en archivo/s **.css** que pueden ser usados en la web

```
$ sass --watch input.scss:output.css
```

VARIABLES

Almacenamos información que vamos a reutilizar a lo largo de nuestra hoja de estilos.

La declaración de variables siempre empieza con \$

```
$mainFont: 'Helvetica', sans;  
$mainColor: #c3c3c3;
```

```
p {  
  font: 10px $mainFont;  
  color: $mainColor;  
}
```

```
p {  
  font: 10px 'Helvetica', sans;  
  color: #c3c3c3; }
```

TIPOS DE VARIABLES

Declaraciones de algunos tipos de variables

- Booleanos

```
$rounded: true;
```

- Números

```
$fontSize: 10;  
$lineHeight: 0.7em;
```

- Colores

```
$main: purple;  
$alert: #ff0000;  
$transparent: rgba(white, 0.5);
```

TIPOS DE VARIABLES (CONT.)

- Literales

```
$font: 'Helvetica';  
$content: "Cargando..."
```

- Listas

```
$colors: red, green, blue, yellow;  
$margins: 10px 40px 0 10px;
```

ÁMBITO DE LAS VARIABLES

Las variables declaradas dentro de una declaración (**dentro de { }**) no pueden ser utilizadas fuera de ese bloque

```
p {  
  $mainFont: 'Helvetica', sans;  
  font: 10px $mainFont;  
}  
h1 { font: 14px $mainFont; }
```

```
error: unbound variable $mainFont
```

ÁMBITO DE LAS VARIABLES (CONT.)

Definiendo nuevos valores a variables "globales", modificamos su valor para el resto de instancias

```
$mainColor: #c3c3c3;  
  
p {  
  color: $mainColor;  
  $mainColor: #333;  
}  
h1 { color: $mainColor; }
```

```
p {color: #c3c3c3; } h1 {color: #333; }
```


ESCAPANDO EL VALOR DE LAS VARIABLES

Podemos escapar el valor de las variables para utilizar como **selector, propiedad o literal** utilizando el formato: **`#{ $variable }`**

```
$top : top;

p {
  span-#{ $top } {
    #{ $top } : 10px;
  }
}
```

```
p span-top {top: 10px; }
```

NESTING

(Selectores enlazados)

Podemos enlazar selectores unos dentro de otros utilizando la misma herarquía que se encuentran en los documentos HTML

```
nav {  
  border-bottom: 1px solid #ff0000;  
  
  ul {  
    list-style: none;  
    li { padding-left: 10px; }  
  }  
  
  a {  
    text-decoration: none;  
    &:hover {  
      color: gray;  
    }  
  }  
}
```

El símbolo & hace referencia al selector padre

NESTING

(Propiedades enlazadas)

Algunas propiedades también pueden enlazarse (font, border, padding, ...)

```
p {  
  &.especial {  
    font: {  
      family: "Helvetica", sans;  
      color: red;  
      weight: bold;  
    }  
  }  
}
```

IMPORT

(Importación de archivos externos)

CSS permite importar porciones de código de otros archivos, haciendo nuestro código mantenible y escalable.

```
@import url('partial/partial.css');
```

Por cada `@import`, el navegador realiza una petición HTTP
Sass importa los archivos combinándolos, sirviendo un sólo archivo CSS

```
@import 'partials/partial';

.container {
  margin:0 auto;
  width: 996px;
  padding: 0 10px;
}
```

en la importación no es necesario incluir la extensión del archivo

PARTIALS

Podemos generar archivos parciales con pequeños trozos de código (snippets) que serán incluidos dentro de el archivo final css

Éstos archivos parciales van nombrados con un **guión bajo** (_) en el nombre del archivo.

`_partials.scss`

```
// _partials.scss  
@import 'partials/partial';
```

Los partials no se compilan como css's

MIXINS

Bloques de código repetitivo que puede tomar argumentos opcionales.
Se comportan como funciones

- **Declaración:** `@mixin mixinName { //codigo }`
- **Llamada:** `@include mixinName;`

```
@mixin linkBtn {  
  text: {  
    decoration:none;  
    transform: uppercase;  
  }  
  color: #fabada;  
}  
  
ul.mainNav {  
  a {  
    @include linkBtn;  
    background-color: #c3c3c3;  
  }  
}
```

@mixin

```
@mixin boxShadow{
  -webkit-box-shadow: 0px 1px 1px rgba(black, 0.3);
  -moz-box-shadow: 0px 1px 1px rgba(black, 0.3);
  -o-box-shadow: 0px 1px 1px rgba(black, 0.3);
  -ms-box-shadow: 0px 1px 1px rgba(black, 0.3);
  box-shadow: 0px 1px 1px rgba(black, 0.3);
}

.module-a {
  @include boxShadow;
  background-color: white;
}

.module-b {
  @include boxShadow;
  background-color: red;
}
```

@mixin (argumentos)

```
@mixin boxShadow($ops){
  -webkit-box-shadow: $ops;
  -moz-box-shadow: $ops;
  -o-box-shadow: $ops;
  -ms-box-shadow: $ops;
  box-shadow: $ops;
}

.module-a {
  @include boxShadow(0px 1px 1px rgba(black, 0.3));
  background-color: white;
}

.module-b {
  @include boxShadow(0px 1px 1px rgba(red, 0.3));
  background-color: red;
}
```


@mixin (arg1, arg2, ..., argN)

```
@mixin boxShadow($xoffset, $yoffset, $blur, $color){  
  -webkit-box-shadow: $xoffset $yoffset $blur $color;  
  -moz-box-shadow: $xoffset $yoffset $blur $color;  
  -o-box-shadow: $xoffset $yoffset $blur $color;  
  -ms-box-shadow: $xoffset $yoffset $blur $color;  
  box-shadow: $xoffset $yoffset $blur $color;  
}  
  
.module-a {  
  @include boxShadow(0px, 1px, 1px, rgba(black, 0.3));  
  background-color: white;  
}  
  
.module-b {  
  @include boxShadow(0px, 1px, 1px, rgba(red, 0.3));  
  background-color: red;  
}
```

HERENCIA

Extensión de clases

@extend .class

```
.alert{
  position: absolute;
  top: 5%; left: 5%; bottom: 5%; right: 5%;
  background-color: white;
  padding: 1em;
  text-align: center;
}

.warning{
  @extend .alert;
  border: 2px solid orange;
}

.success{
  @extend .alert;
  border: 2px solid olive;
}
```

HERENCIA

!! Atencion !!

```
.btn-a{  
  text-transform:uppercase;  
}  
  
.btn-b { @extend .btn-a; }  
  
.container .btn-a{  
  text-transform: uppercase;  
}
```

```
.btn-a, .btn-b {  
  text-transform: uppercase; }  
  
.container .btn-a, .container .btn-b {  
  text-transform: uppercase; }
```

Desde que una **clase extiende a la otra**, todo cambio que se produzca en la clase original, también lo hará en las extendidas, generando código innecesario

HERENCIA

Solución

Placeholder selector (%)

Puede extenderse, pero nunca actúa como un selector en sí mismo

```
%btn{
  text-transform: lowercase;
}
.btn-a { @extend %btn; color: green; }
.btn-b { @extend %btn; color: red; }

.container .btn-a {
  text-transform: uppercase;
}
```

```
.btn-a, .btn-b { text-transform: lowercase; }
.btn-a { color: green; }
.btn-b { color: red; }
.container .btn-a { text-transform: uppercase; }
```

OPERACIONES

Aritmética básica

- + suma
- - resta
- * multiplicación
- / división
- % módulo

```
$family: "Helvetica " + "Neue"; // "Helvetica Neue"  
font-size: 10px + 4pt; // font-size: 14px;  
width: 600px / 960px * 100%; // width: 62.5%;
```

OPERACIONES

Utilidades matemáticas predefinidas

- **round(\$number)** - redondeo
- **ceil(\$number)** - redondeo arriba
- **floor(\$number)** - redondeo abajo
- **abs(\$number)** - valor absoluto
- **min(\$list)** - valor mínimo de una lista
- **max(\$list)** - valor máximo de una lista
- **percentage(\$number)** - valor a porcentaje

```
width: percentage(350px/1000px); // width: 35%;  
line-height: ceil(1.2em); // line-height: 2em;
```

OPERACIONES

Otras funciones predefinidas (colores)

```
$color: #87bf64;  
.grayscale {  
  color: grayscale($color);  
}  
.invert {  
  color: invert($color);  
}  
.complement {  
  color: complement($color);  
}
```

DIRECTIVAS DE CONTROL

- @if ... @else if ... @else
- @each
- @for
- @while

DIRECTIVAS DE CONTROL

@if ... @else if ... @else

```
$theme: pink;

header {
  @if $theme == dark {
    background: #000;
  } @else if $theme == pink {
    background: pink;
  } @else {
    background: #fff;
  }
}
```

DIRECTIVAS DE CONTROL

Operadores de comparacion

- == igual a
- != no es igual / distinto
- > mayor que
- <menor que
- >= mayor ó igual que
- <=menor ó igual que

DIRECTIVAS DE CONTROL

@each

```
$authors: nick aimee dan drew;  
  
@each $author in $authors {  
  .author-#{ $author } {  
    background: url(author-#{ $author }.jpg);  
  }  
}
```

DIRECTIVAS DE CONTROL

@for

```
$authors: nick aimee dan drew;

.item {
  position: absolute;
  right: 0;

  @for $i from 1 through length($authors) {
    $author : nth($authors, $i);
    &.item-#{ $author } {
      top: $i * 30px;
    }
  }
}
```

FRAMEWORK COMPASS



Dispone de una gran cantidad de @mixins para css3 que facilitan la creación de hojas de estilos para los navegadores modernos

<http://compass-style.org/>

THAT'S ALL
FOLKS!!



<http://sass-lang.com/>