

Básico

Variables

```
$red: #8333;
```

```
body {
  color: $red;
}
```

Composing

```
@import './other_sass_file';
@use './other_sass_file';
```

La regla @import está desaconsejada porque acabará siendo eliminada del lenguaje. En su lugar, utiliza la regla @use. La extensión .scss o .sass es opcional.

Mixins

```
@mixin heading-font {
  font-family: sans-serif;
  font-weight: bold;
}
h1 {
  @include heading-font;
}
```

con parámetros

```
@mixin font-size($n) {
  font-size: $n * 1.2em;
}
body {
  @include font-size(2);
}
```

con valores por defecto

```
@mixin pad($n: 10px) {
  padding: $n;
}
body {
  @include pad(15px);
}
```

con una variable por defecto

```
$default-padding: 10px;
establecer un valor por defecto
@mixin pad($n: $default-padding) {
  padding: $n;
}
body {
  @include pad(15px);
}
```

Cada bucle (simple)

```
$menu-items: home about services contact;

@each $item in $menu-items {
  .photo-#{ $item } {
    background: url('images/#{ $item }.jpg');
  }
}
```

Nesting

```
.markdown-body {
  a {
    color: blue;
    &:hover {
      color: red;
    }
  }
}
a propiedades
text: {
  align: center;
  como text-align: center;
  transform: uppercase;
  como text-transform: uppercase;
}
```

Extend

```
.button {
  ...
}
.push-button {
  @extend .button;
}
```

Comentarios

```
/* comentarios en bloque */
// Comentarios de una sola línea
```

Bucles

Para los bucles

```
@for $i from 1 through 4 {
  .item-#{ $i } { left: 20px * $i; }
}
```

Cada bucle (anidado)

```
$backgrounds: (home, 'home.jpg'),
              (about, 'about.jpg');
```

```
@each $id, $image in $backgrounds {
  .photo-#{ $id } {
    background: url($image);
  }
}
```

Bucles while

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

Funciones de color

rgba

```
rgb(100, 120, 140)
rgba(100, 120, 140, .5)
rgba($color, .5)
```

Mixing

```
mix($a, $b, 10%)
```

10% a, 90% b

Obtener valores individuales

HSLA

```
hue($color)
0deg...360deg
saturation($color)
0%...100%
lightness($color)
0%...100%
alpha($color)
0...1 (también como opacity())
```

RGB

```
red($color)
0...255
green($color)
blue($color)
```

Modificación de HSLA

```
darken($color, 5%)
lighten($color, 5%)
saturate($color, 5%)
desaturate($color, 5%)
grayscale($color)
adjust-hue($color, 15deg)
complement($color)
como adjust-hue(_, 180deg)
invert($color)
fade-in($color, .5)
también como opacity()
fade-out($color, .5)
también como transparentize() -
reduce la opacidad a la mitad
rgba($color, .5)
establece el alfa en 0,5
```

Ajustes

Cambios por cantidades fijas

```
adjust-color($color, $blue: 5)
adjust-color($color, $lightness: -30%)
como darken(_, 30%)
adjust-color($color, $alpha: -0.4)
como fade-out(_, .4)
adjust-color($color, $hue: 30deg)
como adjust-hue(_, 15deg)
Cambios mediante porcentajes
scale-color($color, $lightness: 50%)
Cambia una propiedad por completo
```

```
change-color($color, $hue: 180deg)
change-color($color, $blue: 250)
```

Otras características

Condicionales

```
@if $position == 'left' {
  position: absolute;
  left: 0;
}
@else if $position == 'right' {
  position: absolute;
  right: 0;
}
@else {
  position: static;
}
```

Mapas

```
$map: (key1: value1,
key2: value2,
key3: value3);

map-get($map, key1)
```

Interpolación

```
#{ $klass } { ... }
```

Clase

```
call($function-name)
```

Funciones

```
@media #{ $tablet }
font: #{ $size } / #{ $line-height }
url("#{ $background }.jpg")
```

Listas

```
$list: (a b c);

nth($list, 1)
empieza con 1
length($list)

@each $item in $list { ... }
```

Comprobación de características

Comprobar característica

```
feature-exists(*)
```

Características

```
*global-variable-shadowing
*extend-selector-pseudoclass
*units-level-3
*at-error
```

Otras funciones

Cadenas de texto

```
unquote('hello')
quote(hello)
to-upper-case(hello)
to-lower-case(hello)
str-length(hello world)
str-slice(hello, 2, 5)
"ello" - se basa en 1, no en 0
str-insert("abcd", "X", 1)
"Xabcd"
```

Unidades

```
unit(3em)
'em'
unitless(100px)
false
```

Números

```
floor(3.5)
ceil(3.5)
round(3.5)
abs(3.5)
min(1, 2, 3)
max(1, 2, 3)
percentage(.5)
50%
random(3)
0..3
```

Miscelánea

```
variable-exists(red)
comprueba si hay $red
mixin-exists(red-text)
comprueba si @mixin red-text
function-exists(redify)
global-variable-exists(red)
selector-append('.menu', 'li', 'a')
.menu li a
selector-nest('.menu', '&:hover li')
.menu: hover li
selector-extend(...)
selector-parse(...)
selector-replace(...)
selector-unify(...)
```