

Less vs Saas

Less

- Sintaxis: (ejemplo)

```
@base: #f938ab;

.box-shadow(@style, @c) when (iscolor(@c)) {
  -webkit-box-shadow: @style @c;
  -moz-box-shadow:    @style @c;
  box-shadow:        @style @c;
}
.box-shadow(@style, @alpha: 50%) when (isnumber(@alpha)) {
  .box-shadow(@style, rgba(0, 0, 0, @alpha));
}
.box {
  color: saturate(@base, 5%);
  border-color: lighten(@base, 30%);
  div { .box-shadow(0 0 5px, 30%) }
}
```

- Variables

Las variables permiten especificar valores que van a ser utilizados en un único lugar, y además poder reusarlas en otras partes de la hoja de estilo, pudiendo realizar cambios globales modificando una única línea de código.

```
// LESS                                     /* CSS compilado */

@color: #4D926F;

#header {
  color: @color;
}
h2 {
  color: @color;
}

#header {
  color: #4D926F;
}
h2 {
  color: #4D926F;
}
```

- Mixins

Los mixins permiten incluir todas las propiedades de una clase en otra diferente añadiendo simplemente el nombre de la clase como una propiedad más. El comportamiento es similar al de las variables pero con clases enteras. Los mixins pueden comportarse también como funciones, y tomar argumentos, como se ve en el siguiente ejemplo.

```
// LESS                                     /* CSS compilado */

.rounded-corners (@radius: 5px) {
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
  -ms-border-radius: @radius;
  -o-border-radius: @radius;
  border-radius: @radius;
}

#header {
  .rounded-corners;
}
#footer {
  .rounded-corners(10px);
}

#header {
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  -ms-border-radius: 5px;
  -o-border-radius: 5px;
  border-radius: 5px;
}
#footer {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  -ms-border-radius: 10px;
  -o-border-radius: 10px;
  border-radius: 10px;
}
```

- Funciones & Operaciones

¿Existe dependencia entre elementos de tu hoja de estilo? Las operaciones te permiten sumar, restar, multiplicar y dividir los valores de las propiedades y colores, dándote el poder de crear relaciones complejas entre propiedades. Las operaciones deberían estar definidas entre paréntesis para asegurar la compatibilidad con CSS. La función se corresponde a funciones JavaScript, permitiéndote modificar valores como quieras.

```
// LESS                                     /* CSS compilado */

@the-border: 1px;
@base-color: #111;
@red:        #842210;

#header {
  color: (@base-color * 3);
  border-left: @the-border;
  border-right: (@the-border * 2);
}
#footer {
  color: (@base-color + #003300);
  border-color: desaturate(@red, 10%);
}

#header {
  color: #333;
  border-left: 1px;
  border-right: 2px;
}
#footer {
  color: #114411;
  border-color: #7d2717;
}
```

SASS

- Sintaxis y variable:

```
// creando variable
$red: #FF0000;

body {
  // aplicando el valor de $red
  background-color: $red;
}
```

- Mixins

Los mixins se definen con la directiva @mixin seguida del nombre del mixin (y opcionalmente una lista de argumentos) y seguida por el bloque de contenidos que definen los estilos del mixin. El siguiente ejemplo define un mixin sin argumentos llamado large-text:

```
@mixin large-text {
  font: {
    family: Arial;
    size: 20px;
    weight: bold;
  }
  color: #ff0000;
}
```

- Funciones

Al margen de las funciones propias definidas por Sass, también es posible definir funciones propias para que puedas utilizarlas en tus hojas de estilos. Ejemplo:

```
$grid-width: 40px;
$gutter-width: 10px;

@function grid-width($n) {
  @return $n * $grid-width + ($n - 1) * $gutter-width;
}

#sidebar { width: grid-width(5); }
```

Se vería compilado así:

```
#sidebar {  
  width: 240px;  
}
```

- Cálculos

En SASS no puedes hacer eso por dos razones:

1. No puedes hacer cálculos en tiempo real, SASS compila y te devuelve un archivo CSS, no tienes acceso al HTML desde SASS.
2. No puedes mezclar unidades distintas en las operaciones matemáticas en SASS.

Sin embargo puedes hacer algo en CSS con la función `calc()`. Esta función te permite hacer cálculos en tiempo real y manejar diferentes unidades.

```
.contenedor{  
  min-height: calc(100vh - 50px );  
  background: #ccc;  
  border:1px solid  
}
```