

## Unidad 4: SASS – parte 2

2º DAW

Curso 2022/2023

Mari Cruz Gómez García

## ÍNDICE

6. MÓDULOS

7. CONDICIONALES

8. BUCLES

9. MEDIA QUERIES

## 6. MÓDULOS

Modularizar es crear múltiples archivos y en vez de tener todo en una única hoja de estilo, los vamos a separar en distintos archivos y se unen todos en una de la siguiente forma:



3

## 6. MÓDULOS

Para que el compilador de Sass no compile un módulo por separado, tiene que saber qué archivo es un módulo. Para ello simplemente vamos a nombrar el módulo comenzando por un guión bajo "\_".

Otro gran consejo a la hora de definir tus módulos es aplicar la "componetización". Los estilos de cada pequeño componente de tu interfaz web deben colocarse en un archivo aparte.

4

## 6. MÓDULOS

Por ejemplo, podemos tener un módulo para definición de cosas como:

- Botones
- Capas modales
- Paneles
- Tooltips
- Breadcrumbs (migas de pan)
- Cabecera
- Pie...etc

Así, cada vez que quieras modificar cualquier pequeña parte de la interfaz de tu sitio, sabrás en qué archivo CSS están esos estilos. Encontrarás rápidamente las reglas CSS a modificar y ayudarás a otras personas que más adelante tengan que lidiar con tu código CSS.

5

## 7. CONDICIONALES: if

Sabemos que la sentencia if corresponde a una estructura condicional.

Hay varios operadores para realizar una comparativa en Sass:

- Igual → ==
- No es igual → !=
- Mayor que → >
- Mayor o igual que → >=
- Menor que → <
- Menor o igual que → <=

Debemos tener en cuenta que estos cuatro últimos (<, <=, >, >=) son sólo para comparar números.

VER EJEMPLO CONDICIONALES.HTML

```
1 // Condicionales
2 // TEORIA
3 $color: purpura;
4
5 h1{
6   @if ($color == azul) {
7     color: blue;
8   }
9   @else if ($color == verde){
10    color: green;
11  }
12  @else{
13    color: red;
14  }
15 }
```

6

## 8. BUCLES: each

La directiva @each nos permite iterar una lista por cada uno de los ítems que tiene.

Para iterar cada uno de los ítems de la lista, comenzaremos usando el bucle each:

@each \$item in \$lista:

### application.scss

```
$autores: rosa lucia pablo jose;
@each $autor in $autores {
  .author-#{$autor}{
    background: url(author-#{$autor}.jpg);
  }
}
```

¿Cuál es el resultado css?

7

## 8. BUCLES: each

### application.css

```
.author-rosa{
  background: url(author-rosa.jpg);
}

.author-lucia{
  background: url(author-rosa.jpg);
}
.author-pablo{
  background: url(author-rosa.jpg);
}
.author-jose{
  background: url(author-rosa.jpg);
}
```

8

## 8. BUCLES: for

Estas sentencias funcionan como el @each pero a diferencia del anterior se encuentra asociado por lo general a números. Por ejemplo, si queremos realizar una iteración del 1 al 3 de forma que para cada uno de los valores del bucle (1..3) lo multiplicaremos por 30 pixeles de forma que cuando compilemos tengamos una lista de ítems que use los i valores:

```
application.scss
.item{
  position: absolute;
  right: 0;
  @for $i from 1 through 3{
    &.item-#{$i}{
      top: $i * 30px;
    }
  }
}
```

¿Cuál es la salida css?

9

## 8. BUCLES: for

```
application.css

.item{
  position: absolute;
  right: 0;
}
.item.item-1{
  top: 30px;
}
.item.item-2{
  top: 60px;
}
.item.item-3{
  top: 90px;
}
```

10

## 8. BUCLES: while

Algo parecido sería si en lugar de usar el @for usamos el @while, la diferencia radica en que con el @while debemos ir actualizando el índice en cada iteración.

¿cómo se quedaría el ejemplo anterior?

11

## 8. BUCLES: while

### application.scss

```
$i: 1;
.item{
  position: absolute;
  right: 0;
  @while $i < 4{
    &.item-#{ $i }{
      top: $i * 30px;
    }
    $i: $i+1;
  }
}
```

12

## 9. MEDIA QUERIES

Antes en CSS:

```
@media screen and (max-width: 900px) {  
  p{  
    font-size: 60px;  
    color: crimson;  
  }  
}  
  
@media screen and (max-width: 700px) {  
  p{  
    font-size: 40px;  
    color: greenyellow;  
  }  
}  
  
@media screen and (max-width: 400px) {  
  p{  
    font-size: 20px;  
    color: #000;  
  }  
}
```

13

## 9. MEDIA QUERIES

Lo normal en media queries es que nosotros establezcamos los tamaños para desktop , para tablets y para smartphones.

**RECUERDA:** En SASS #{} es una interpolación.

Con ella se puede usar una variable en SASS en un Selector o identificador.

```
$desktop:1200px;  
$tablets:600px;  
$phones:480px;  
  
@mixin desktop {  
  @media screen and (max-width: #{ $desktop} ) {  
    @content;  
  }  
}  
  
@mixin tablets {  
  @media screen and (max-width: #{ $tablets} ) {  
    @content;  
  }  
}  
  
@mixin phones {  
  @media screen and (max-width: #{ $phones} ) {  
    @content;  
  }  
}
```

14

## 9. MEDIA QUERIES

Estamos creando un media-query para párrafos:

¿Qué hace @content?

Escribe y prueba este Código.

```
$desktop:1200px;  
$tablets:600px;  
$phones:480px;
```

```
@mixin desktop {  
  @media screen and (max-width: #{ $desktop} ) {  
    @content;  
  }  
}
```

```
@mixin tablets {  
  @media screen and (max-width: #{ $tablets} ) {  
    @content;  
  }  
}
```

```
@mixin phones {  
  @media screen and (max-width: #{ $phones} ) {  
    @content;  
  }  
}
```

```
p{  
  font-size: 80px;  
  color: lightseagreen;  
  
  @include desktop(){  
    font-size: 60px;  
    color: crimson;  
  }  
  
  @include tablets(){  
    font-size: 40px;  
    color: greenyellow;  
  }  
  
  @include phones(){  
    font-size: 20px;  
    color: #000;  
  }  
}
```