
Sass

SASS

- SASS (hoja de estilo sintácticamente impresionante), es un preprocesador de CSS, más estable y potente que el lenguaje de extensión CSS.
- http://sass-lang.com/

- SASS permite escribir CSS más fácilmente. Se pueden hacer más cosas, con menos código, más legible y en menos tiempo.

Características

- Más estable potente y compatible con versiones de CSS.
- Es un súper conjunto de CSS y se basa en JavaScript.
- Utiliza su propia sintaxis y compila a CSS legible.
- Es posible escribir CSS en menos código y en menos tiempo.
- Utiliza métodos reutilizables, instrucciones lógicas e incorpora algunas funciones como la manipulación de color, matemáticas y listas de parámetros.

Instalando SASS

Instalación

- **Standalone (Recomendado)**
  - Download desde GitHub: https://github.com/sass/dart-sass/releases/tag/1.17.0
  - Añadir al PATH.
  - No hay dependencias.
- **Desde Node.js**
  - npm install -g sass
- **En Mac OS X (Homebrew)**
  - brew install sass/sass/sass

Creando un proyecto Sass

Crea el HTML

- Crea una carpeta de proyecto y añade un index.html

```
<html>
  <head>
    <title> Import example of sass</title>
    <link rel = "stylesheet" type = "text/css" href = "style.css"/>
  </head>

  <body>
    <h1>Primer ejemplo</h1>
    <h3>Welcome to Sass</h3>
  </body>
</html>
```

Crea el archivo SASS

- Crear el archivo “style.scss”, que tendrá una estructura similar a la del archivo CSS, solo que con la extensión .**scss**.

```
h1{
  color: #AF80ED;
}

h3{
  color: #DE5E85;
}
```

Compilar SASS

- Abre una ventana de comando en la carpeta de proyecto
- Ejecuta la orden

```
sass --watch style.scss:style.css
```

- Esta orden estará pendiente de los cambios en el archivo style.scss y generará automáticamente el archivo “style.css”

```
D:\sass\ejemplo1> sass --watch style.scss:style.css
>>> Sass is watching for changes. Press Ctrl-C to stop.
      write style.css
      write style.css.map
>>> Change detected to: style.scss
      write style.css
      write style.css.map
```

- Para terminar el comando pulsa **CTRL+C**

Comprueba el resultado

- Abre el archivo index.html en un navegador

Variables

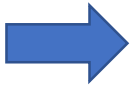
- Almacenan información para reusarla en la hoja de estilo. Ej.: colores, fuentes o cualquier valor de CSS.
- Se representan con el símbolo $:

```
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

Reglas anidadas

- Es una forma de combinar varias reglas CSS dentro de otra.
  - SCSS

```scss
.container{
  h1{
    font-size: 25px;
    color:#E45456;
  }

  p{
    font-size: 25px;
    color:#3C7949;
  }

  .box{
    h1{
      font-size: 25px;
      color:#E45456;
    }

    p{
      font-size: 25px;
      color:#3C7949;
    }
  }
}
```



  - CSS

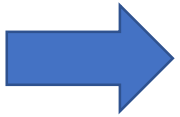
```css
.container h1 {
  font-size: 25px;
  color: #E45456;
}

.container p {
  font-size: 25px;
  color: #3C7949;
}

.container .box h1 {
  font-size: 25px;
  color: #E45456;
}

.container .box p {
  font-size: 25px;
  color: #3C7949;
}
```

Propiedades anidadas

- Permite anidamiento de propiedades en otras propiedades que conducen a la agrupación de otro código relacionado.

- SCSS

```
.line {
  font: {
    family: Lucida Sans Unicode;
    size:20px;
    weight: bold;
    variant: small-caps;
  }
}
```



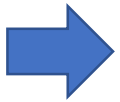
- CSS

```
.line {
  font-family: Lucida Sans
Unicode;
  font-size: 20px;
  font-weight: bold;
  font-variant: small-caps;
}
```

Selectores padre: &

- Se puede seleccionar el selector padre usando el carácter &. Indica dónde se debe insertar el selector padre.
  - SCSS

```
a {
  font-size: 20px;
  &:hover { background-color:
yellow; }
}
```



  - CSS

```
a {
  font-size: 20px;
}
a:hover {
  background-color: yellow;
}
```

Parciales

- Fragmentos de css que se pueden incluir en otros ficheros sass.
- Sirven para modularizar CSS y hacerlo más mantenible
- Requiere nombrar con "_" delante del nombre para que sass lo reconozca como parte de otro fichero css
- Se importan en otros ficheros mediante la directiva:

  @import

Import de parciales

- La directiva **@import** ya existe en CSS para dividir un fichero en fragmentos. Pero provoca un HTTPRequest por cada fragmento.
- En sass las directivas sirven para montar un **único fichero** CSS con todos los fragmentos.

Directivas Mixins

- "**funciones**" de SASS
- Los mixins permite crear un grupo de estilos, que se pueden reutilizar en toda la hoja de estilos.
- Pueden recibir parámetros y proporcionar diferentes salidas.
- Se definen con la directiva **@mixin** seguido del nombre del mixin y llaves que engloban el código
- EJ:

```
@mixin color-invertido {
        background-color: #111;
        color: #eee;
}
```

- Para invocarlo se utiliza la directiva **@include** dentro de la regla donde se quiere utilizar. Ej:

```
h1.invertido {
 font-size: 1.3em;
 padding: 15px;
@include color-invertido;
}
```

Paso de parámetros en un mixin

- Los parámetros permiten proporcionar salidas distintas a un mixin.
- Útil para pasar los prefijos de vendors para algunas propiedades
  - Ej.: Para la función transformar:

```
@mixin transformar($propiedad) {
 -webkit-transform: $propiedad;
 -ms-transform: $propiedad;
 transform: $propiedad;
}
```

- Uso:

```
.escalada {
          @include transformar(scale(2, 3)) }

h1 {
          color: blue;
          @include transformar(rotate(22deg))

          }
```

Paso de valores de propiedades en un mixin

- Ej.

```
@mixin encabezados($tamano) {
        h1 {
                font-size: $tamano; }
        h2 {
                font-size: $tamano - 0.2; }
        h3 {
                font-size: $tamano - 0.5; }
}
```

- Uso en mediaqueries:

```
@include encabezados(1.5em);

        @media(min-width: 800px) {
                @include encabezados(2em);
        }
        @media(min-width: 1200px) {
                @include encabezados(2.5em);
        }
```

Selector placeholder y herencia

- Se definen con el símbolo **"%"** delante del nombre de la clase y se utilizan para obtener un conjunto de reglas base que luego pueden extenderse en los hijos.

```
%heading {
  background-color: blanchedalmond;
  color: brown;
  font-family: 'Times New Roman', Times, serif;
}
```

- Uso;

```
h1 {
        @extend %heading;
        font-size: 2em;
}
h2 {
        @extend %heading;
        font-size: 1.5em; }
```

Comentarios

- SASS es compatible con dos tipos de comentarios:
- Comentarios multilinea:
  - Estos comentarios se escriben con "/*" y "*/". Los comentarios multilineas se conservan en la salida de CSS.
- Comentarios de una línea:
  - Estos se escriben utilizando "//" seguido del comentario. Los comentarios de una línea no se conservan en la salida CSS.

```
/* This comment is
 * more than one line long
 * since it uses the CSS comment syntax,
 * it will appear in the CSS output. */
body { color: black; }

// These comments are in single line
// They will not appear in the CSS output,
// since they use the single-line comment syntax.
a { color: blue; }
```

```
/* This comment is
 * more than one line long
 * since it uses the CSS comment syntax,
 * it will appear in the CSS output. */
body {
  color: black; }

a {
  color: blue; }
```

Ejercicio

- Genera el scss de los distintos tipos de cajas que puede tener tu página:
  - Caja base: padding: 10px; font-size: 1.2em;
  - Caja con borde: Caja + border: 1px solid #ddd;
  - Caja con fondo: Caja + background-color: #f0f0f0;
  - Caja con espaciado extra: Caja + padding: 20px;
  - Caja combinada: Caja + borde + fondo + espaciado extra
- Utiliza Asignación al padre (&) y herencia (%) para formatear distintos

Directivas de función

- En SASS se puede crear una función propia y utilizarla en el contexto del script o con cualquier valor.
- Las funciones se llaman usando el nombre de la función, con sus parámetros.

Usando directiva de funciones

El HTML

- Creamos el index.html

```
<html>

<head>
    <title>Nested Rules</title>
    <link rel = "stylesheet" type = "text/css" href = "style.css" />
</head>

  <body>

    <div class = "container" id = "set_width">
      <h2>Example for Function directives</h2>
      <p>SASS stands for Syntactically Awesome Stylesheet. </p>
    </div>

 </body>

</html>
```

El .scss

- Definimos el archivo style.scss

```
$first-width: 5px;
$second-width: 5px;

@function adjust_width($n) {
   @return $n * $first-width + ($n - 1) * $second-
width;
}

#set_width { padding-left: adjust_width(10); }
```

Compilamos

- Cada vez que se modifique el archivo .scss, se debe ejecutar el comando:

```
sass --watch style.scss:style.css
```

- Se generará o actualizará automáticamente el archivo .css:

```
#set_width {
  padding-left: 95px;
}
```

- Comprobamos abriendo el index.html en el navegador

Estilos de Salida

- El archivo CSS que SASS genera tiene el estilo de CSS por defecto, que se refleja en la estructura del documento.
- La salida por defecto es buena, pero en ocasiones podría no ser adecuada para todas las situaciones, por lo que SASS es compatible con otros estilos de salida.
- Los estilos disponibles son:
  - :nested
  - :expanded
  - :compact
  - :compressed
- La orden para definir la salida se invoca de la siguiente manera:

```
sass --watch style.scss:style.css --style compressed
```

Estilos de Salida: fuente scss

- Consideremos el siguiente código Sass para observar los distintos tipos de salida

```
.widget-social {
    text-align: right;

    a,
    a:visited {
        padding: 0 3px;
        color: #222222;
        color: rgba(34, 34, 34, 0.77);
     }

    a:hover {
        color: #B00909;
    }

}
```

:nested

- Es el estilo por defecto de SASS.
- Es muy útil cuando el archivo CSS es demasiado grande, esto hace que la estructura del archivo sea mas legible y pueda ser comprendido fácilmente.
- Por ejemplo:

```
.widget-social {
  text-align: right; }
  .widget-social a,
  .widget-social a:visited {
    padding: 0 3px;
    color: #222222;
    color: rgba(34, 34, 34, 0.77); }
  .widget-social a:hover {
    color: #B00909; }
```

:expanded

- Se necesita mas espacio en comparación con estilo anidado.
- La sección de reglas consiste en propiedades, las cuales están todas dentro de las reglas, mientras que las reglas no sigan sumando sangría.
- Ejemplo:

```
.widget-social {
  text-align: right;
}
.widget-social a,
.widget-social a:visited {
  padding: 0 3px;
  color: #222222;
  color: rgba(34, 34, 34, 0.77);
}
.widget-social a:hover {
  color: #B00909;
}
```

:compact

- Ocupa menos espacio que las otras formas.
- Se centra principalmente en los selectores (en vez de en sus propiedades)
- Cada selector y sus propiedades se colocan en la misma salida.
- Ejemplo:

```
.widget-social { text-align: right; }
.widget-social a, .widget-social a:visited { padding: 0 3px; color: #222222; color: rgba(34, 34, 34, 0.77); }
.widget-social a:hover { color: #B00909; }
```

:compressed

- Ocupa el menor espacio en comparación con los otros estilos mencionados anteriormente.
- Proporciona espacios en blanco solo a los selectores y nueva línea al final del archivo.
- Este forma de salida es confusa y no es fácil de leer.
- Ejemplo

```
.widget-social{text-align:right}.widget-social a,.widget-social a:visited{padding:0 3px;color:#222222;color:rgba(34,34,34,0.77)}.widget-social a:hover{color:#B00909}
```