

SASS&BS



INSTALACIÓN Y USO

SASS ES UN PREPROCESADOR CSS

Un preprocesador de CSS es un lenguaje de programación de hoja de estilos

Nos permite utilizar funcionalidades como funciones, variables, anidaciones y demás, lo que nos permite definir CSS con un lenguaje más cercano a la programación.

Todos los navegadores actuales pueden trabajar con CSS, pero no con SCSS (~~e-SASS~~), razón por la cual es necesario procesarlos antes de incorporarlos al sitio web.

Dicho de otro modo, el código del documento pre-procesado (SASS, SCSS, LESS) debe ser convertido a CSS puro antes de realizar el despliegue del sitio web.



SCSS		CSS
<pre>section {</pre>	1	<pre>section {</pre>
<pre> height: 100px;</pre>	2	<pre> height: 100px;</pre>
<pre> width: 100px;</pre>	3	<pre> width: 100px;</pre>
<pre>}</pre>	4	<pre>}</pre>
<pre>.class-one {</pre>	5	
<pre> height: 50px;</pre>	6	<pre>section .class-one {</pre>
<pre> width: 50px;</pre>	7	<pre> height: 50px;</pre>
<pre>}</pre>	8	<pre> width: 50px;</pre>
<pre>.button {</pre>	9	<pre>}</pre>
<pre> color: #074e68;</pre>	10	
<pre>}</pre>	11	<pre>section .class-one .button {</pre>
<pre>}</pre>	12	<pre> color: #074e68;</pre>
<pre>}</pre>	13	<pre>}</pre>

SASS: *SYNTACTICALLY AWESOME STYLE SHEET*



SASS es el más conocido de los pre-procesadores CSS, es de código abierto y está disponible bajo licencia MIT.

[https://sass-lang.com.](https://sass-lang.com)

El uso SASS o cualquier otro preprocesador nos provee de diferentes funcionalidades no incluidas en CSS puro y que harán el trabajo con CSS más sencillo, así como la reutilización de código en diferentes proyectos.

- Variables
- Funciones matemáticas
- Estructuras lógicas y de control
- Mixins o plantillas
- Indentaciones
- Anidamientos
- Herencias
- Ficheros parciales
- Etc.

```

$fuente: Helvetica, sans-serif;

$color: #F00;

$color_menu: #FFF;

$color_enlaces: #000;

$fondo: #CCC;

html{
    background-color: $fondo;
}

body{
    font: 100% $fuente;
    color: $color;
}

nav{
    ul{
        margin: 0;
        padding: 0;
        list-style: none;
        color: $color_menu;
    }
    li{
        display: inline-block;
    }
    a{
        display: block;
        padding: 6px 12px;
        text-decoration: none;
        color: $color_enlaces;
    }
}

```

SCSS



```

html {
    background-color: #CCC;
}

body {
    font: 100% Helvetica, sans-serif;
    color: #F00; }

nav ul {
    margin: 0;
    padding: 0;
    list-style: none;
    color: #FFF; }

nav li {
    display: inline-block; }

nav a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
    color: #000; }

```

CSS

INSTALACIÓN

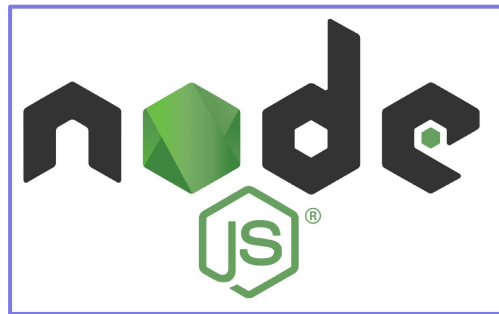
Hay diferentes formas de instalar SASS, como por ejemplo a través de instaladores como CodeKit, Compass y similares, pero es mejor hacer de forma manual, tanto en entornos Linux, como Windows.

- Lo primero será comprobar si tenemos node.js instalado con el comando `node -v`

```
PowerShell 7 (x64)
PS C:\Users\ccarnero> node -v
v20.11.0
PS C:\Users\ccarnero>
```

Un preprocesador de CSS es un lenguaje de programación d

- Si no lo tenemos instalado, o tenemos una versión anterior a V.12, debemos instalar la última versión descargando el instalador de Node.js desde su web:



INSTALACIÓN

- Una vez instalado Node.js, ya podemos instalar SASS desde consola con el comando `npm install -g sass`

```
PS C:\Users\carlo> npm install -g sass
C:\Users\carlo\AppData\Roaming\npm\sass -> C:\Users\carlo\AppData\Roaming\npm\sass
npm WARN SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.3.2 (node_modules\fsevents):
npm WARN SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@~2.3.2:
  {"os": "win32", "arch": "x64"}
+ sass@1.71.1
added 3 packages from 39 contributors, removed 121 packages and updated 1 package in 1.711s
PS C:\Users\carlo>
```

Si durante el proceso sale uno o varios mensajes indicando algún error, debemos seguir las instrucciones indicadas.

USO DE SASS: SINTAXIS

VARIABLES

Las variables en SASS se declaran con el símbolo dólar, \$, antes del nombre.

El valor se le asigna usando el símbolo 2 puntos :

\$nombre_var: valor;

ATENCIÓN A LOS IDENTIFICADORES CON GUIONES.

SASS no diferencia entre guiones
bajos y medios.

```
$color-menu: #f87732;

$margin-ul: .2rem;

ul{

    margin: $margin_ul;

    padding: $margin_ul;

    list-style: none;

    color: $color_menu;

}
```


COMENTARIOS

Los comentarios en SCSS funcionan de forma similar a los de JavaScript:

- Comentarios de una línea:
`// esto es un coment. 1 línea`
 - Van hasta el final de la línea
 - No son procesados en el CSS de salida (silent comments).
- Comentarios multi-línea:
`/* este es
multilínea */`
 - El comentario aparecerá en CSS.

```
// Este comentario no  
aparecerá en CSS.
```

```
/* Este comentario sí,  
salvo que el CSS esté  
comprimido. */
```

NESTING



```
nav{  
  ul{  
    margin: 0;  
    padding: 0;  
    list-style: none;  
    color: $color_menu;  
  }  
  li{  
    display: inline-block;  
  }  
  a{  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
    color: $color_enlaces;  
  }  
}
```

SCSS

CSS

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
  color: #FFF;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
  color: #000;  
}
```

COMPILAR SCSS EN CSS

COMPILACIÓN

Para crear un fichero CSS a partir del fichero .scss, este debe ser compilado.

- De manera general se compila el fichero CSS de forma manual mediante consola:

```
> sass ruta/ficheroSass.scss:ruta/ficheroCss.css
```

- Hay muchos editores que compilan de forma automática el CSS cuando se detectan modificaciones en el fichero original SCSS.

```
# compilar style.scss en style.css.
```

```
$ sass style.scss:style.css
```

```
# compilar light.scss y dark.scss a light.css y dark.css.
```

```
$ sass light.scss:light.css dark.scss:dark.css
```

```
# Compila todos los ficheros de la carpeta themes/ a ficheros CSS en  
public/css/.
```

```
$ sass themes:public/css
```

PERSONALIZACIÓN DE BOOTSTRAP CON SASS

PERSONALIZACIÓN BOOTSTRAP CON SASS

Ya sabemos que una de las desventajas del uso de BS es que todos los websites se parecen.

Con el uso de SASS podemos modificar cualquier parámetro (variable) de BS a nuestro gusto.

La forma más o menos estandarizada de personalización es crear archivo SCSS de personalización, (normalmente de nombre ***scss/custom.scss***) y dentro de él proceder con la personalización:

- Importar recursos: fuentes, librerías de iconos, etc
- Definir variables propias.
- Modificar variables de BS.
- Importar fichero SCSS de BS.

PERSONALIZACIÓN BOOTSTRAP CON SASS

```
// Custom.scss
```

```
// 1. Include functions first (so you can manipulate colors, SVGs, calc, etc)
```

```
@import "../node_modules/bootstrap/scss/functions";
```

```
// 2. Include any default variable overrides here
```

```
// 3. Include remainder of required Bootstrap stylesheets (including any separate color mode stylesheets)
```

```
@import "../node_modules/bootstrap/scss/variables";
```

```
@import "../node_modules/bootstrap/scss/variables-dark";
```

```
// 4. Include any default map overrides here
```

```
// 5. Import all the rest of Bootstrap
```

```
@import "../node_modules/bootstrap/scss/bootstrap";
```

```
// Then add additional custom code here
```

<https://www.initialapps.com/customizing-bootstrap-5-with-sass/>

PERSONALIZACIÓN BOOTSTRAP CON SASS

scss/_variables.scss

```
$theme-colors: (  
  "primary":    $primary,  
  "secondary":  $secondary,  
  "success":    $success,  
  "info":       $info,  
  "warning":    $warning,  
  "danger":     $danger,  
  "light":      $light,  
  "dark":       $dark  
);
```

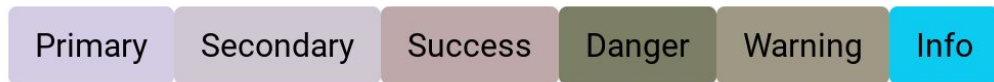
```
// 2. Include any default variable overrides here
```

```
$primary: #D4CBE5;  
$secondary: #CFC7D2;  
$success: #BEA8AA;  
$warning: #9E9885;  
$danger: #7C7F65;
```

Bootstrap Buttons!

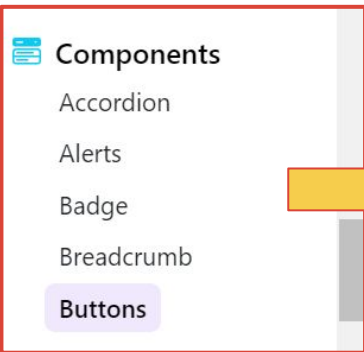


Bootstrap Buttons!



PERSONALIZACIÓN BOOTSTRAP CON SASS

Para conocer las variables de BS que deseamos modificar, no tenemos más que dirigirnos a la sección del elemento de BS que queremos modificar, en la documentación y ver su apartado de variables SASS.



Sass variables

scss/_variables.scss

```
$btn-color: var(--#{$prefix}body-color);  
$btn-padding-y: $input-btn-padding-y;  
$btn-padding-x: $input-btn-padding-x;  
$btn-font-family: $input-btn-font-family;  
$btn-font-size: $input-btn-font-size;  
$btn-line-height: $input-btn-line-height;
```

```
// PERSONALIZACIÓN BOTONES BS  
$btn-font-family: $tipo-roboto;  
$btn-font-size: 1.5rem;
```

FIN