

Preprocesadores

Módulo 03

Implementación SASS

Implementación

SASS

Al ser un **preprocesador** me permite mantener herramientas **de programación** para nuestros estilos.

En principio las **principales ventajas de trabajar con preprocesadores**, es ampliar las capacidades del lenguaje.

Por ejemplo nos permite **modularizar así como trabajar con variables globales, operadores y elementos** que facilitan la implementación **especialmente en grandes sitios**.



Implementación

SASS

Es importante saber qué aprender **SASS** también nos facilitará el trabajo posterior con frameworks de CSS muy utilizados como **Bootstrap, Bulma o Materialize**.

Para saber más sobre el uso específico de cada uno recomendamos **ir al apartado de sass que cada uno posee en sus páginas oficiales,**

- <https://materializecss.com/sass.html>
- <https://bulma.io/documentation/customize/with-node-sass/>
- <https://getbootstrap.com/docs/5.0/customize/sass/>



Implementación

SASS

Para comenzar a trabajar con **SASS**, necesitamos compilarlo.

¿Por qué necesitamos compilarlo?

Básicamente **porque al ser un preprocesador no está realizado en CSS.**

Para poder compilarlo necesitamos un compilador. Tenemos diversas opciones para poder realizarlo.

Una opción es a través **MPM**, por terminal con **node.js**.



Implementación

SASS

Para poder comenzar a trabajar con **node.js**, podemos ir a su **página oficial (nodejs.org y descargarlo)**

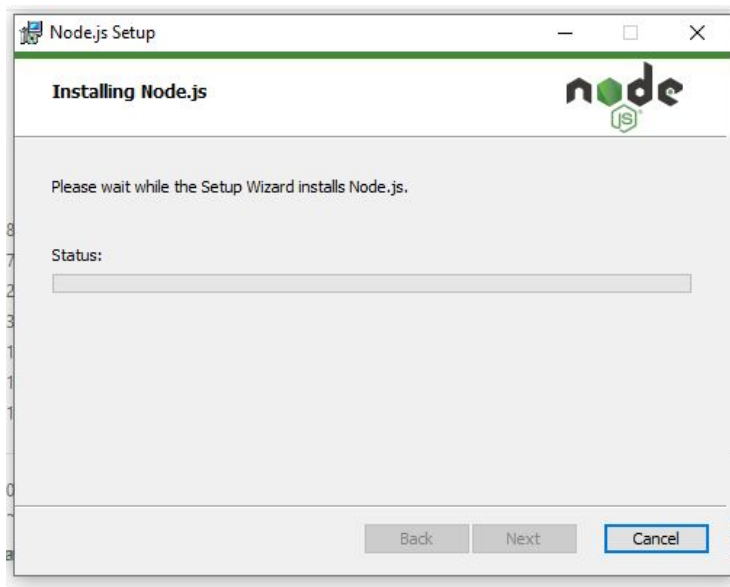


A screenshot of the Node.js website homepage. At the top is the Node.js logo (a green hexagon with 'node' and 'js' in white). Below the logo is a dark navigation bar with links: INICIO, ACERCA, DESCARGAS, DOCUMENTACIÓN, PARTICIPE, SEGURIDAD, NOTICIAS, and CERTIFICATION. The main content area has a dark background with the text: "Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome." Below this is a dark button with the text "#BlackLivesMatter". Further down is a light green banner with the text "New security releases to be made available January 4, 2021". Below the banner are two green buttons: "14.15.3 LTS" with the subtext "Recomendado para la mayoría" and "15.4.0 Actual" with the subtext "Últimas características". At the bottom are two links: "Otras Descargas | Cambios | Documentación del API" repeated twice. At the very bottom is the text "O eche un vistazo a la Programa de soporte a largo plazo (LTS)."

Implementación

SASS

Como comenzamos la instalación en la [slide anterior](#), ahora simplemente debemos seguir un **wizard muy sencillo e instalarlo**,



Implementación

SASS

Sin embargo, para poder utilizar la opción anterior, si bien sencilla luego de trabajar con **node.js**, nos obliga a conocer el lenguaje con la terminal correspondiente.

Por supuesto que no es necesario aprender node.js para poder trabajar con **SASS**. Sólo mencionamos esa opción porque para quien luego de trabajar con **node.js** quiera a su vez integrar sus conocimientos de sass debe saber qué puede manejar ambos procesos a través de la **misma terminal**.



04 Símbolo del sistema

```
Microsoft Windows [Versión 10.0.19041.685]  
(c) 2020 Microsoft Corporation. Todos los derechos reservados.  
C:\Users\
```


Implementación

SASS

Por esa razón una opción bastante interesante es el trabajo con **www.prepos.io**

Esta página es muy fácil de utilizar, y nos permite descargar un programa que **permitirá la compilación de forma sencilla y sin conocimientos adicionales,**

Your Friendly Web Development Companion

Prepos compiles your files, transpiles your JavaScript, reloads your browsers and makes it really easy to develop & test your websites so you can focus on making them perfect.

Download Free Unlimited Trial

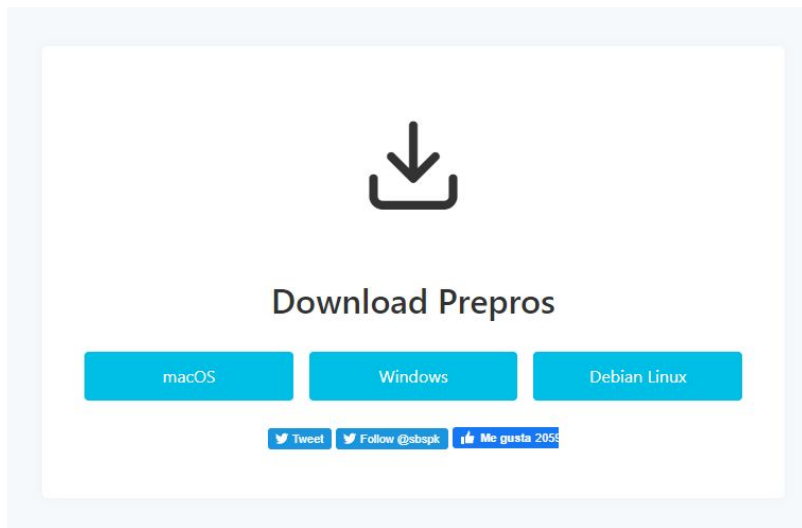
Available for Windows, macOS and Linux
Trial includes all Prepos features.



Implementación

SASS

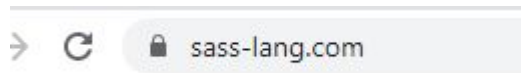
Luego elegir la opción adecuada según **nuestro sistema operativo**,



Implementación

¿Cómo empezar a trabajar?

El [paso anterior](#) es una de las múltiples opciones en la página oficial de **SASS** podemos encontrar una lista de posibles compiladores aunque siempre es bueno chequear en las páginas oficiales de **los mismos características y compatibilidad con versiones**,



Si luego de ingresar a la página hacemos [click en Install](#).



Implementación

sass

Luego de [ingresar a la página de SASS](#) podemos luego de hacer click en la opción **Install**, obtener una lista de compiladores u opciones posibles,



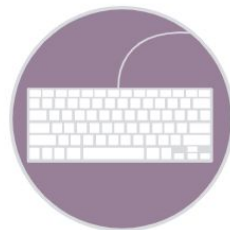
Applications



There are a good many applications that will get you up and running with Sass in a few minutes for Mac, Windows, and Linux. You can download most of the applications for free but a few of them are paid apps (and totally worth it).

- CodeKit (Paid) Mac
- Ghostlab (Paid) Mac Windows
- Hammer (Paid) Mac
- LiveReload (Paid, Open Source) Mac Windows
- Prepros (Paid) Mac Windows Linux
- Scout-App (Free, Open Source) Windows Linux Mac

Command Line



When you install Sass on the command line, you'll be able to run the `sass` executable to compile `.sass` and `.scss` files to `.css` files. For example:

```
sass source/stylesheets/index.scss build/stylesheets/index.css
```

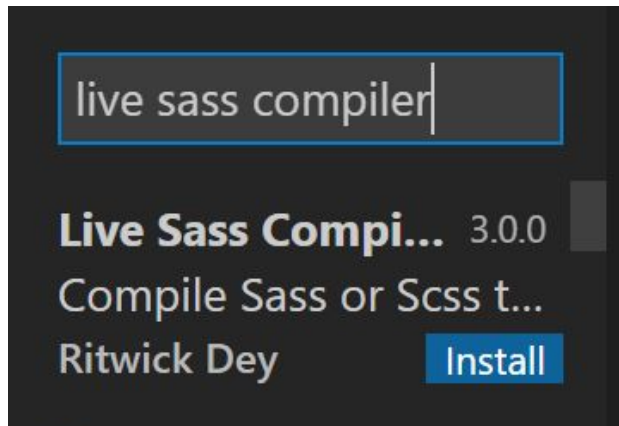
First install Sass using one of the options below, then run `sass --version` to be sure it installed correctly. If it did, this will include 1.30.0. You can also run `sass --help` for more information about the command-line interface.

Once it's all set up, go and play. If you're brand new to Sass we've set up some resources to help you learn pretty darn quick.

Implementación

sass

Más allá de todas las opciones válidas anteriores, que dejan gran libertad de elección, y **en definitiva no son más que una forma de compilarlo**, ya que la parte importante es aprender a trabajar con el lenguaje, **nosotros en nuestro caso vamos a elegir un complemento de Visual Code**,



Implementación

sass

Empecemos a trabajar!

Lo primero es **generar nuestro primer archivo**. Todos los archivos deben tener la extensión **scss** **qué es la propia de los archivos sass**,



Nombre de archivo:	<input type="text" value="estilos.scss"/>
Tipo:	<input type="text" value="Plain Text"/>

La pregunta es , qué archivo vamos a vincular **con nuestro archivo.html**

Implementación

sass

Como vimos en la [diapositiva anterior](#), hemos guardado nuestro archivo **estilos.scss** y nuestro archivo **index.html**.

Pero, la vinculación no será directa, pues el **.scss** no es el archivo que puede vincularse a .html, entonces vinculamos una archivo que aún no existe pero pronto aparecerá,



```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Práctica SASS</title>
<link rel="stylesheet" href="css/estilos.css">
```

Implementación

sass

Empecemos a trabajar!

Ahora generemos una regla de estilo cómo usualmente hacemos,

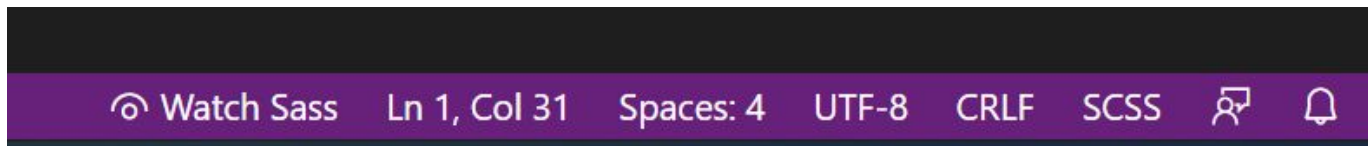


```
<> index.html X  estilos.scss X
C: > Users > sacab > OneDrive > Escritorio > cosas > practicaSass > css > estilos.scss > body
1  body { background-color: blue; }
```

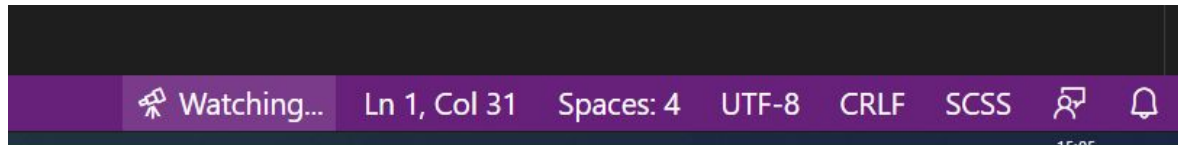

Implementación

sass

Al momento de utilizar nuestro compilador de complemento, vamos a notar que **debajo en visual code aparece lo siguiente,**



Al hacerle click a watch Sass, cambia a



Implementación

sass

Empecemos a trabajar!

Gracias al paso anterior veremos qué se ha generado un **archivo .css que justamente** es aquel vinculado con nuestro **index.html**



```
# estilos.css  
# estilos.css.map  
🔗 estilos.scss  
<> index.html
```

Implementación

sass

Una de las principales ventajas de **sass** es el trabajo con variables,

```
estilos.scss X
css > estilos.scss > body
1  $color : blue;
2  body { background-color: $color; }
```



Siempre la variable se declara de la siguiente forma,

```
estilos.scss X
css > estilos.scss > body
1  $variable : valor;
```

Implementación

sass

Empecemos a trabajar!

También podemos **trabajar con operadores**, por ejemplo



```
estilos.scss X
css > estilos.scss > ...
1  $color : blue;
2  $ancho1 : 300px;
3  $mitad: $ancho1 / 2;
4
5
6  body { background-color: $color;}
7  main { width: $ancho1;}
8  section { width: $mitad; }
```

Implementación

sass

Empecemos a trabajar!

El resultado del **código anterior** será,

```
estilos.scss # estilos.css X
css > # estilos.css > body
1 body {
2   background-color: blue;
3 }
4
5 main {
6   width: 300px;
7 }
8
9 section {
10  width: 150px;
11 }
12 /*# sourceMappingURL=estilos.css.map */
```



Implementación

sass

Una forma usual **de trabajo** es utilizar en nuestro .css selectores **descendentes**,

```
h1 { text-decoration:underline; }  
h1 span { color: gray; }
```

Sin embargo, en este caso, lo **que haremos** será trabajar de la **misma manera** pero desde nuestro sass.

Facilitará **el mantenimiento** y es poderosamente más intuitivo,



Implementación

sass

Para **generar** exactamente [lo anterior](#) pero trabajando sass utilizamos la **anidación**,

```
h1 {  
  text-decoration: underline;  
  span { color: gray; }  
}
```



Implementación

sass

Al momento de **compilarlo el resultado será,**



A screenshot of a file explorer interface. It shows two files: 'estilos.scss' with a red icon and a close button (X), and 'estilos.css' with a blue icon.

El siguiente,

```
h1 {  
  text-decoration: underline;  
}  
  
h1 span {  
  color: gray;  
}  
  
/*# sourceMappingURL=estilos.css.map */
```



Implementación

sass

Pero la anidación en sass puede ser más interesante aún utilizando selectores padres como el siguiente,

```
.seccion {  
  @extend %seccion;  
  
  &-especial{  
    @extend %seccion;  
    @extend %seccion-descripcion;  
  }  
  
  &-descripcion {  
    @extend %seccion;  
    @extend %seccion-descripcion;  
  }  
  
  &-conjunto {  
    @extend %seccion;  
    @extend %seccion-especial;  
    @extend %seccion-descripcion;  
  }  
}
```



Implementación

sass

De esta forma el resultado será,

```
index.html estilos.scss # estilos.css x
css > # estilos.css > .seccion
1 v .seccion, .seccion-especial, .seccion-descripcion, .seccion-conjunto {
2   padding: 10px;
3   font-size: 1.2em;
4   color: green;
5   font-family: Verdana, Geneva, Tahoma, sans-serif;
6   width: 100%;
7 }
8
9 v .seccion-conjunto {
10  border: 1px solid #ddd;
11 }
12
13 v .seccion-especial, .seccion-descripcion, .seccion-conjunto {
14  background-color: #b4d43e;
15 }
16 /*# sourceMappingURL=estilos.css.map */
```



Revisión

- Repasar los conceptos vistos de **Javascript con audio y video**
- Trabajar con el **bonus track** debajo de todo para practicar los **elementos individualmente**.
- Ver todos los videos y materiales necesarios antes de continuar



¡Muchas gracias!

¡Sigamos trabajando!