

Que es SASS

SASS es un preprocesador de código CSS, es decir, todo lo que escribas acabará siendo código CSS, ya que es lo que los navegadores pueden entender. Lo bueno de SASS es que puedes extender el código CSS para hacerlo más reutilizable y dinámico, en otras palabras, con SASS agilizas el desarrollo y consigues escribir menos código CSS



Que es SASS

SASS utiliza sus propios ficheros, y en ese caso tiene dos vertientes:

Ficheros .sass: En estos ficheros no necesitas escribir llaves ni punto y coma al final. Si has utilizado alguna vez Stylus, Python o alguno de esos lenguajes te vas a sentir más cómodo con estos archivos.

Ficheros .scss: En esta vertiente si que puedes escribir llaves y puntos y coma al final, por lo tanto puedes escribir código CSS dentro de estos ficheros como hacías hasta ahora.

Ventajas de SASS

La principal ventaja de SASS es la posibilidad de convertir los CSS en algo dinámico. Permite trabajar mucho más rápido en la creación de código con la posibilidad de crear funciones que realicen ciertas operaciones matemáticas y reutilizar código gracias a los mixins, variables que nos permiten guardar valores. SASS, en definitiva, se convierte en tu mejor ayudante.

Y algo siempre importante cuando te decantas por una herramienta con alternativas, dispone de una gran comunidad que la hace progresar, por lo que se le augura un gran futuro por delante.

Como utilizar SASS

Para usar SASS necesitas un compilador que transforme tus ficheros .scss o .sass en archivos .css cotidianos.

Lo más fácil es instalar SASS mediante su paquete global de NPM.

```
npm install -g sass
```

Como utilizar SASS

Para usar SASS con la consola o terminal de tu sistema operativo o incluso en el mismo visual studio podemos escribir unos comandos para poder compilar por ejemplo:
Los archivos input serán los archivos que utilizaremos para usar sass y los output serán los archivos de salida que interpretara el navegador. Existen dos extensiones .Sass .Scss

```
sass --watch input.scss output.css
```

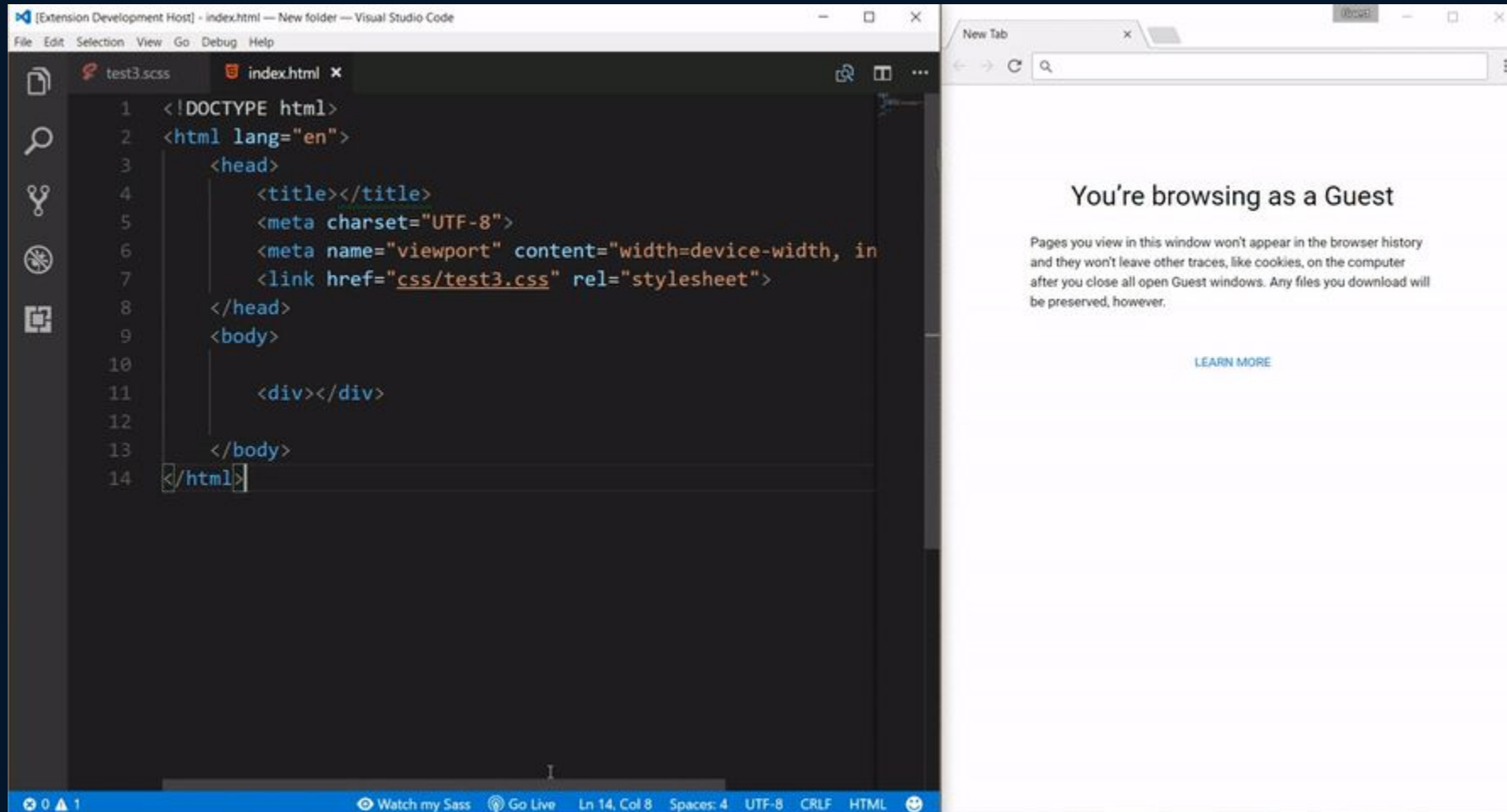
```
sass --watch input.sass output.css
```

```
sass --watch carpeta/input.scss output.css
```

Como utilizar SASS

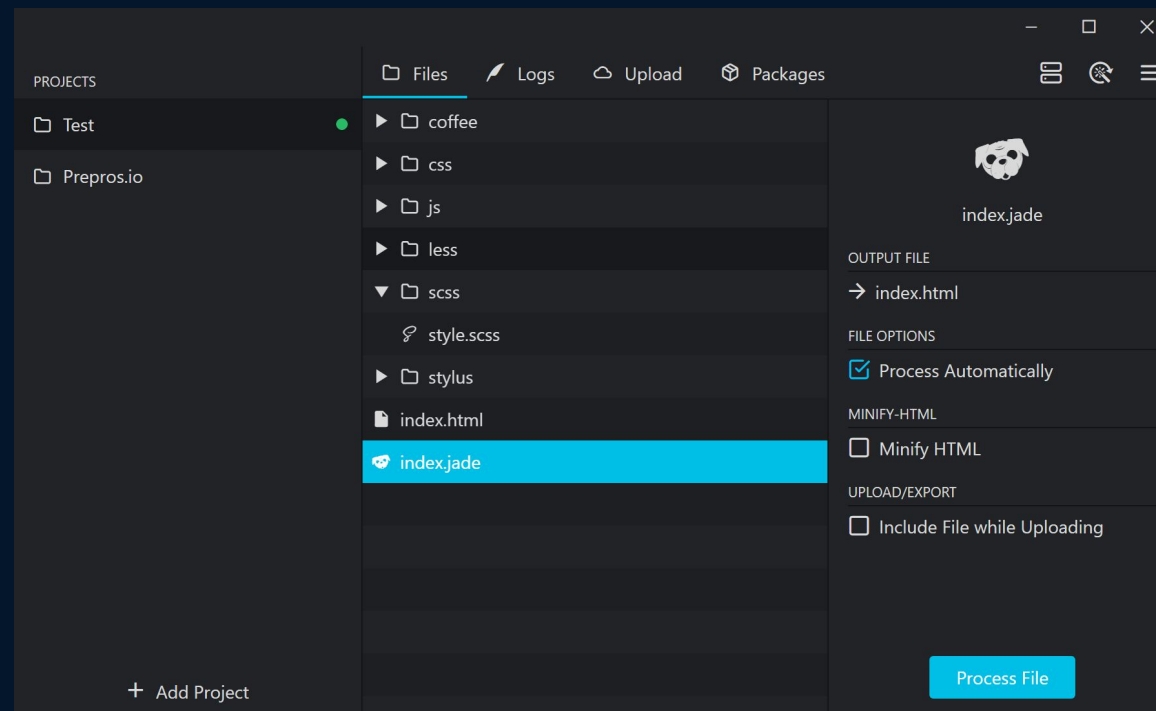
Para usar SASS en Visual Studio Code podemos utilizar una extensión llamada live sass compiler que trabaja en conjunto con la extensión live server. Con esta extensión podemos compilar nuestro código de SASS a un archivo de CSS solo con hacer un clic en el botón watch sass

Como utilizar SASS



Como utilizar SASS

Para usar SASS también podemos descargar algún software que compile a CSS, existen varios programas que permiten compilar varios preprocesadores no solo en CSS sino también en otros lenguajes por ejemplo Prepros Koala Scout-app



Programas

<https://prepros.io/>
<http://koala-app.com/>
<https://scout-app.io/>

Sintaxis de SASS

Podemos usar dos extensiones y cada extensión tiene su propia sintaxis Scss es igual a CSS pero SASS no utiliza llaves y tampoco se cierran las propiedades con ;

```
1  SASS
2
3  .head
4    font-size: 12px;
5    color: blue;
6    .head-top
7      font-family: arial
8
9  SCSS
10
11  .head {
12    font-size: 12px;
13    color: blue;
14    .head-top {
15      font-family: arial;
16    }
17  }
18
```

Anidación o Nesting en SASS

Una de las mejoras que ofrece sass es que permite anidar elementos dentro de otros elementos. Por ejemplo

CSS

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

SCSS

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

Variables en SASS

Sass permite la incorporación de variables solo debemos declararla por medio de un signo pesos(\$) delante del nombre a declarar

```
$gray-color:  #888888;  
$white-color:  #FFFFFF;  
$red-color:  red;  
$yellow-color:  yellow;  
$blue-color:  blue;  
  
.header {  
    background-color: $gray-color;  
    color: $white-color;  
}
```

Variables en SASS

Pero antes que es una variable?

Con una variable podemos almacenar un dato solo debemos utilizar el signo pesos(\$) y darle un valor.

\$mivariable = mivalor

Tipos de datos

- ✓ Números (ejemplo: 1.2, 13, 10px)
- ✓ Cadenas de texto con o sin comillas simples o dobles (ejemplo "foo", 'bar', baz)
- ✓ Colores (ejemplo blue, #04a3f9, rgba(255, 0, 0, 0.5))
- ✓ Valores lógicos o booleanos (ejemplo true, false)
- ✓ Valores nulos (ejemplo null)
- ✓ Listas de valores, separados por espacios en blanco o comas (ejemplo 1.5em 1em 0 2em, Helvetica, Arial, sans-serif)
- ✓ Pares formados por una clave y un valor separados por : (ejemplo (key1: value1, key2: value2))

Variables en SASS

Tipos de variables

Existen dos tipos de variables las locales y las globales.

La única diferencia entre una variable local y una global es su ámbito de acción.

Variable local: es aquella que sólo es utilizable dentro del código de la función.

Variable global: es aquella cuyo ámbito es todo el programa, incluso dentro de una función.

Dicho de otro modo:

Una variable declarada dentro de una función es una variable de tipo local que sólo existe dentro del contexto de la función. Por contrapartida, una variable global tiene un ámbito de todo el programa (incluso dentro de las funciones).

Variables con valores por defecto SASS

Variables

La palabra reservada !default permite controlar la asignación de valores a las variables de manera mucho más precisa. Si una variable ya tenía un valor asignado, !default hace que se mantenga sin cambios. Si la variable no existía o no tenía ningún valor, se utiliza el nuevo valor asignado. Ejemplo:

```
$contenido: "Primer contenido";  
$contenido: "¿Segundo contenido?" !default;  
$nuevo_contenido: "Tercer contenido" !default;  
  
#main {  
  contenido: $contenido;  
  nuevo-contenido: $nuevo_contenido;  
}
```

Operadores en SASS

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

```
//suma en sass
#imagen{
  height: 4px + 2;
}
```

```
//Resta en sass
#imagen{
  height: 4px - 2;
}
```

```
//multiplicación en sass
#imagen{
  height: 4px * 2;
}
```

```
//Division en sass
#imagen{
  height: 4px / 2;
}
```

La regla @import

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

```
//multiplicación en sass
#imagen{
  height: 4px * 2;
}
```


Los condicionales @if

Los condicionales

Esta directiva se utiliza por medio del @if para evaluar una condición.

Si la condición no se cumple y es falsa(FALSE) no se aplican estilos.

Si la condición se cumple y es verdadera(TRUE) se aplican los estilos a determinado selector.

Los condicionales @if

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

@for

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

@each

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

maps

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

@while

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

La regla @extend

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

mixins

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

@include

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible:

Functions

Tipos de operaciones matemáticas en sass

SassScript soporta los cinco operadores aritméticos básicos: suma +, resta -, multiplicación *, división / y módulo %. El operador módulo calcula el resto de la división sin decimales (ejemplo: 5 módulo 2 = 1, $5 \% 2 = 1$). Además, si realizas operaciones sobre números con diferentes unidades, Sass convertirá automáticamente las unidades siempre que sea posible: