

Sass

¿Qué es Sass?

- ▶ Sass o Syntactically Awesome Stylesheets, es un preprocesador CSS, que fue desarrollado inicialmente en el año 2006 por Hampton Catlin, Natalie Weizenbaum y Chris Eppstein.
- ▶ Tiene un lenguaje de hojas de estilo, su código es open source y está escrito en Ruby. Por eso, para poder instalar y utilizar Sass , previamente debemos tener instalado Ruby.

Preprocesadores CSS

Un preprocesador CSS es una herramienta que sirve para que los desarrolladores de sitios web escriban código CSS más fácil y limpio. Tiene su propio código de hojas de estilo no estándar, que posteriormente se compila en un código CSS estándar, ya que los navegadores no entienden la sintaxis.

Existen diferentes preprocesadores como Sass, Less y Stylus.



Todos los procesadores, nos ofrecen más o menos las mismas utilidades, pero sin duda el que destaca sobretodos los demás es Sass, ya que se ha convertido en el más usado y demandado.

Al haber recibido un mayor apoyo de la comunidad de desarrolladores, es probable encontrar ofertas de trabajo con Sass o incluso heredar proyectos que usan Sass.

Además, varios frameworks usan Sass, como es el caso de bootstrap.



Ventajas de usar Sass

- ▶ Al desarrollar con un preprocesador se consigue principalmente un ahorro de tiempo, ya que tenemos que escribir menos código para hacer las cosas.
- ▶ También conseguimos una mayor facilidad de mantenimiento del código, gracias a una mejor organización del código y la posibilidad de editar una vez ciertos valores y que afecten a decenas, o cientos, de lugares del código CSS generado.



Dos tipos de sintaxis:

HTML

Sass

Result

```
$verde: #33D374
$blanco: #FFF

div
  background-color: $verde
  padding: 10px 0 10px
  text-align: center
  color: $blanco
```

Resources

View Compiled

HTML

SCSS

Result

```
$verde: #33D374;
$blanco: #FFF;

div {
  background-color: $verde ;
  padding: 10px 0 10px;
  text-align: center;
  color: $blanco;
}
```

Resources

View Compiled

Características:



Variables

Sass utiliza el símbolo \$ para reconocer una variable, la cual permite almacenar valores (como colores, tamaños, tipos de fuente, etc.) y reutilizarlos más tarde.

```
$variable-name: value;
```

```
// SCSS
$font-primary: 'Open Sans', sans-serif;
$primary-color: #333;

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

// SASS
$font-primary: 'Open Sans', sans-serif
$primary-color: #333

body
  font: 100% $font-primary
  color: $primary-color

// CSS
body {
  font: 100% 'Open Sans', sans-serif;
  color: #333;
}
```


Nesting

Sass permite anidar los selectores y propiedades de CSS y así crear una jerarquía visual similar a la que se ve en HTML. Aplicar buenas prácticas en cuanto al nesting del código facilita su entendimiento, desarrollo y mantenimiento.

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
    color: black;  
  
    &:hover {  
      cursor: pointer;  
      color: red;  
    }  
  }  
}
```

Mixins

Los mixins son funciones Sass que agrupan declaraciones CSS juntas, para así evitar escribir de manera repetitiva el mismo código una y otra vez.

Podemos crear un mixin con el comando **@mixin**, seguido de un nombre:

```
@mixin my-font {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 16px;  
  font-style: italic;  
}
```

```
$font-color: red;  
  
@mixin my-font($font-color) {  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 16px;  
  font-style: italic;  
  color: $font-color;  
}
```

```
p {  
  @include my-font;  
}
```

Importaciones

Podemos dividir nuestros archivos Sass en partes pequeñas. Es mucho más fácil leer y mantener archivos más pequeños en lugar de un archivo grande con líneas interminables. Este tipo de archivo es conocido como parcial.

Posteriormente, podemos importar nuestro archivo Sass parcial con el comando `@import` a un archivo Sass maestro, el cual será utilizado para la traducción de Sass a CSS.

```
1 body,  
2 html,  
3 ol,  
4 ul {  
5     margin: 0;  
6     padding: 0;  
7 }
```

```
1 $txt-size: 16px;  
2 $txt-color: #333;
```

```
1 @import 'reset';  
2 @import 'vars';  
3  
4 body {  
5     color: $txt-color;  
6     font-size: $txt-size;  
7 }
```

```
body, html, ol, ul {  
    margin: 0;  
    padding: 0;  
}  
  
body {  
    color: #333;  
    font-size: 16px;  
}
```

```
$Sass = $time - $a-lot|
```

Sass

