**Curso de sass**

En el mercado existen varios preprocesadores, pero en el día de hoy abordaremos sass, su objetivo como el del resto de preprocesadores es optimizar y simplificar el código hecho en css.

**Ventajas:** la comunidad es mucho más amplia en sass, ya que es un proyecto en de código abierto.

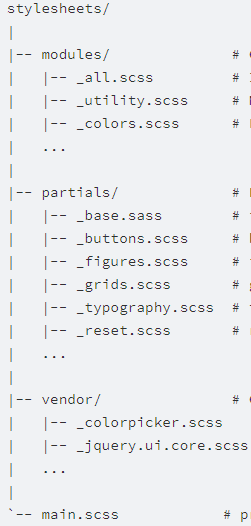
**Video #05: Estructura de css en un archivo sass.**

**Organización por archivos:** una de las ventajas más importantes que poseen en general los preprocesadores, es la organización y separación de nuestro código css por deferentes documentos, pues podemos hacer un documento para elemento de nuestra página, por ejemplo: para nuestro header, main, footer, uno que contenga nuestras variables, todos estos documentos al final serán importados en un solo documento principal que es el cual será compilado a código css.

No está por demás aclarar que este documento principal, seguirá teniendo el reglamento principal de css, que es, la ejecución del código será en cascada.

Para realizar las importaciones a nuestro css principal, es con la palabra reservada ‘**@import**’.

Cada archivo de scss o sass que tendrá el prefijo ‘\_’, por ejemplo, \_nombre. scss.

Esto es válido execto en el archivo principal.

Esta imagen del lado izquierdo contendrá la estructura básica de como tener un proyecto de front con código en sass.

**Modules:** El directorio modules está reservado para el código Sass que no hace que Sass genere CSS realmente. Cosas como declaraciones, funciones y variables de mixin.

**Partials:** este directorio contendrá todos los archivos de los componentes o elementos de dentro archivo HTML.

**Vendor:** Este directorio contendrá todo el en css realizado por tercero, como librerías o demás.

Por ultimo y no menos importante esta en nuestra imagen nuestro archivo principal de scss, este contendrá todas las importaciones y es el archivo que será convertido a css puro.

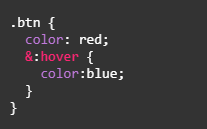
Para más información entra aquí: <http://thesassway.com/beginner/how-to-structure-a-sass-project>

**Video #06: variables.**

Las variables son una forma de almacenar la información que se desea reutilizar a lo largo de la hoja de estilo.

Se puede almacenar cosas como colores, pilas de fuentes o cualquier valor de CSS que se desea reutilizar. Sass usa el símbolo $ para nombrar una variable.

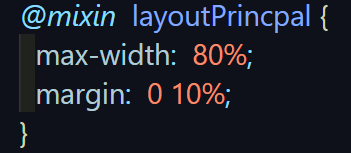
**Nota Video#09:** las anidaciones en pocas palabras, es trabajar con un elemento, o clase que este dentro de nuestra clase padre, por ejemplo, si nuestra clase padre se llama **btn**, dentro de ella hay un icono, bien podemos hacer referencia a ella dentro de nuestra clase padre, o también trabajar con pseudoclases



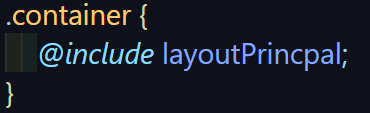
**Video# 10: mixin.**

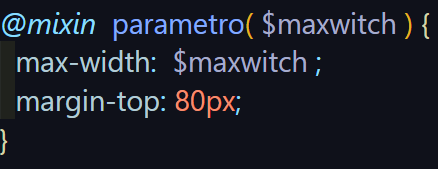
con esta característica, no permite reutilizar código css que es muy repetido dentro de nuestra hoja de estilos.

Por ejemplo, definir nuestro tamaño global en un mixin con su margen.

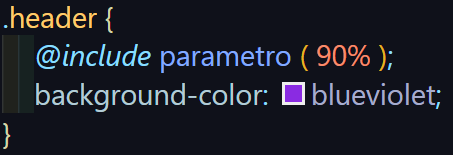


Para declarara un mixin, este debe ir con el prefijo **@mixin** <nombre> y para hace su invocación dentro de una clase se hace de la siguiente manera

con el prefijo **@include** <nombre> dentro de clase que desees.

Dentro de los mixin existen los mixin paramétricos que como su nombre lo indica, reciben un parámetro atreves de su llamada en @**include.** Ejemplo:

es de esta forma que se define un mixin paramétrico.

de esta forma se hace el envío del parámetro hacia el mixin.

Esta directiva también posee otra característica que es el **@content,** este nos permite incluir bloques de código dentro de la comunicación del mixin la invocación del mismo en la clase que deseemos. Ejemplo

