**SASS**

Texto

Descripción generada automáticamente

Es un preprocesador.

Trabajar sobre una herramienta y que esa herramienta que convierte en CSS. Esto nos permite a generar un código CSS mucho más limpio.

Casi que nos permite trabajar con pseudocodigo.

Texto

Descripción generada automáticamente con confianza baja

Texto, Carta

Descripción generada automáticamente

Permite particionar el css en componentes.

Generar el código CSS mucho más rápido.

Hay 3 maneras de instalar SASS:

* Mas compleja
* Media
* Más sencilla

Mientras más compleja es la instalación, más control se tiene sobre la herramienta SASS.

1. Instalar node.js

Node.js es básicamente el entorno de ejecución javascript que nos va a permitir ejecutar javascript por fuera del navegador.

JavaScript funciona nativamente dentro de un navegador. Para trabajarlo localmente descargamos node.js

1. Crear una carpeta nueva (porque SASS elimina todo el CSS del proyecto)
2. Abrimos terminal (Git Bash o Node.js)
3. Npm: Node Package Managment

Es el gestor de paquetes que viene por defecto de Node. Nos facilita la gestión de dependencias y paquetes de software dentro de los proyectos desarrollados con Node.

1. **npm init**

Texto

Descripción generada automáticamente

package.json es la parte fundamental del proyecto.

El “npm init” solo se hace una vez. Así se forma el .json (solo debe a ver uno por proyecto).

1. Vamos dando “ENTER” hasta que se terminen las preguntas

Texto

Descripción generada automáticamente

1. Entramos al proyecto y aparece un archivo package.json

Captura de pantalla de un celular

Descripción generada automáticamente

1. Si sale que hay una actualización pendiente, escribes el comando que te dice y se actualiza

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media



1. **npm install -D node-sass nodemon**

**OTRA FORMA DE INSTALACIÓN:**

1. Creamos el proyecto (con una carpeta para css y scss)
2. Abrimos la terminal en el proyecto
3. “npm init” (ENTER a todo hasta que aparezca el archivo .json)
4. “npm install -g sass”
5. “sass --watch ./scss/style.scss ./css/style.css” o “sass --watch scss:css”

Le indicamos que archivo scss y css son los que vamos a enlazar.

1. Volver a abrir el proyecto
2. Abrir el proyecto en la terminal
3. sass --watch ./scss/style.scss ./css/style.css

Volvemos a iniciar el SASS

SCSS no lo entiende el HTML, por eso se sigue linkeando el css. Solo el SASS entiende el SCSS por eso es un pre-proceso. Antes de CSS.

SASS trabaja con “PARTIALS”

Los partials son pequeños fragmentos de código SASS.

Guarda fragemtnos de nuestro proyecto.

El sitio lo podemos ordenar en 7 carpetas, pero usamos 4 principalmente.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

**Base:** información general. Tipografias, Reset, Estilos Generales

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Para que se entienda que hablamos de partials, se pone el “\_” al inicio del nombre.

**Components:** todo lo que sean componentes (cards de productos, barra de navegación, componentes de Bootstrap)

Texto

Descripción generada automáticamente

**Layout:** todo loque corresponde a los bloques o estructura del código (header, main, , footer, section, article, etc.) -> Etiquetas de primer orden

Pantalla negra con letras blancas

Descripción generada automáticamente

Acá se colocan los Media Query

**Utilities:** todo lo que corresponde a “programación” (variables, funciones, maps, etc.)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

RECORDAR: CSS lee de arriba hacia abajo. Por lo que es importante el orden.

Texto

Descripción generada automáticamente

**EXTENDS (@extend)**

Si quiero que otra “class” tenga las mismas propiedades que otra clase. Sirve para reciclar el código.

También se pueden modificar los valores de las propiedades.

Captura de pantalla de computadora

Descripción generada automáticamente

DATO: Cuando trabajamos con MQ los **@extend** no funcionan. No podemos hacer un @extend de una propiedad que este fuera del Media Query.

**MIXING**

Sirve para agrupar ciertas propiedades.

Para declararlo

**@mixin “nombre\_mixing”{**

**propiedades**

**}**

**@mixin “nombre\_mixing”($parametro1, $parametro2){**

**propiedades**

**}**

Para llamarlo

**@include “nombre\_mixing”($argumento1, $argumento2);**

El argumento puede ser un valor o una variable, cualquiera. Aunque en el curso de Desarrollo Web se utiliza normalmente ingresando el valor directamente.

Texto

Descripción generada automáticamente

TRUCO: “none” en un argumento, indica que esa propiedad no se va a usar.