EDUCACIÓN IT

Bootcamp de Full Stack

Bienvenidos a la clase N°23

Planilla de seguimiento: https://docs.google.com/spreadsheets/d/133uLofSPwbTbMnfN6sL7czACQfvR3GXb7t1M7ND5dTs/edit#gid=0

EDUCACIÓN IT

SASS

Entorno

- **Sass** es un lenguaje de hojas de estilo compilado en CSS . Le permite utilizar variables , reglas anidadas , mixins , funciones y más, todo con una sintaxis totalmente compatible con CSS . Sass ayuda a mantener bien organizadas las hojas de estilo grandes y facilita compartir diseños dentro y entre proyectos.

  Fuente: https://sass-lang.com/documentation/

- Existen varias formas de compilar en Sass, las más conocidas son:
  a. Por medio de NodeJS: sass scss/main.scss css/main.css
  b. A través de VSC instalando la extensión “Live Sass Compiler”.
     i. Hay que configurar el directorio de salida. Esto se hace agregando a settings.json la siguiente línea: "liveSassCompile.settings.formats": [{"format":"expanded", "extensionName":".css", "savePath":"~/../css/"}]
     ii. Por último, se debe ejecutar “Watch Sass” cada vez que se quiera compilar o colocar en modo monitor.

EDUCACIÓN IT

SASS

Variables y Mixins

- Las variables permiten almacenar un dato dentro de ellas. Las misma, tienen alcance de accesibilidad y depende del lugar en donde se declaren. Se las puede declarar como privadas para el módulo.
    - Declaración global: $mi-variable: 10px;
    - Declaración local: .mi-clase { $mi-variable: 50px; }
    - Declaración privada (se coloca un guión bajo después del símbolo $): $_mi-variable: 25px;
    - Declaración variable de múltiples valores: $mi-variable: (25px, 50px, blue); se invoca con nth($mi-variable, 1);
- La regla **@mixin** es empleada para reutilizar un conjunto de declaraciones de estilo en múltiples lugares de tu hoja de estilo. Los mixins permites el uso de parámetros. La utilización del mismo en un contexto dado, se hace por medio de la regla @include.
    - Declaración @mixin square($size, $radius: 0) { propiedades }
    - Utilización @include square(100px, 5px);

EDUCACIÓN IT

SASS

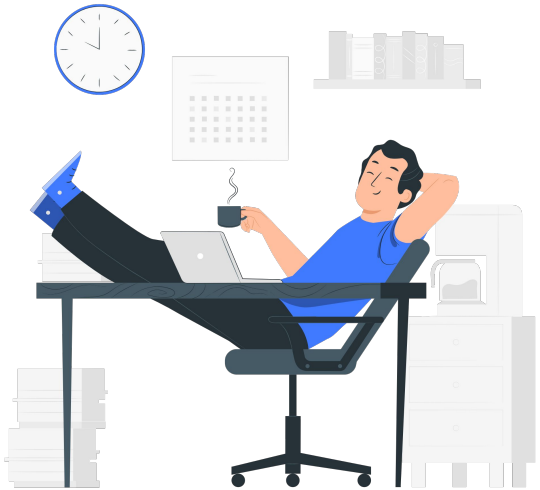
Modularización

- La regla **@use** se utiliza para importar variables, mixins y funciones de diferentes archivos SASS con el fin de componer un archivo CSS. Los archivos importados por @use se denominan "módulos" a los cuales se le puede colocar un alias . Utilización `@use "./carpeta/archivo-sass" as alias;`
- La regla **@forward** se utiliza para re-exportar las variables, mixins y funciones de un módulo a otros módulos, el objetivo es construir una biblioteca que facilita la organización y la reutilización del código. Utilización `@forward "./carpeta/archivo-sass";`
- En resumen, ambas reglas son seguras, controladas y evitan la duplicación de código. La regla @use se utiliza para importar un módulo por completo, mientras que @forward se utiliza para exportar partes específicas de un módulo.

EDUCACIÓN IT

BREAK

Descansemos 15 minutos

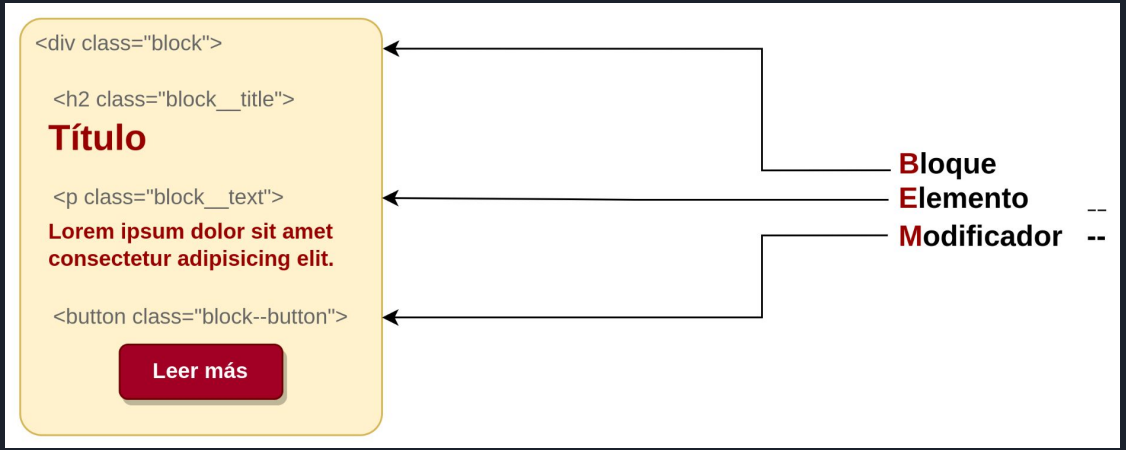


EDUCACIÓN IT

SASS

Nomenclatura BEM

BEM es una abreviatura de los elementos clave: Bloque, Elemento y Modificador. Sugiere una manera estructurada de nombrar nuestras clases, basado en las propiedades del elemento en cuestión. En cuanto al modificador, se refiere a una bandera en un bloque o elemento que se emplea para cambiar su apariencia o comportamiento.



EDUCACIÓN IT

Desafío En Grupo

Sass

Entre todos resolvamos el siguiente ejercicio:

1. Crear un archivos index.html y main.scss para implementar Sass.
2. En el index.html, agregar un contenedor que tenga un título, un párrafo y una imagen de fondo.
3. Crear al menos tres partials “_variables.scss”, “mixins.scss” y “_home.scss”.
4. Emplear variables, mixins y use.
5. Trabajar bajo la nomenclatura BEM.

Tiempo estimado: 25 minutos.

EDUCACIÓN IT

CIERRE DE CLASE

Continuaremos en la próxima clase