Índice:

[Índice: 1](#_Toc133332577)

[¿Qué es Sass? 2](#_Toc133332578)

[Compilar Sass 3](#_Toc133332579)

[Instalar plugin en visual studio code para SASS 4](#_Toc133332580)

[Variables 5](#_Toc133332581)

[Anidamiento 5](#_Toc133332582)

[Tipo de datos: 6](#_Toc133332583)

[@import 7](#_Toc133332584)

[@extend 7](#_Toc133332585)

[Condicionales 8](#_Toc133332586)

[Ciclo @for 9](#_Toc133332587)

[Ciclo @each 9](#_Toc133332588)

[@mixin 9](#_Toc133332589)

[Definición de mixins 9](#_Toc133332590)

[Parámetros y argumentos 10](#_Toc133332591)

[Parámetros múltiples 11](#_Toc133332592)

[Directiva content 12](#_Toc133332593)

[Funciones 13](#_Toc133332594)

[Funciones de strings 13](#_Toc133332595)

[Funciones de números 14](#_Toc133332596)

[Funciones de listas y mapas 14](#_Toc133332597)

[Funciones de introspección 14](#_Toc133332598)

[Color 14](#_Toc133332599)

[Color sustractivo (pinturas, imprentas) 14](#_Toc133332600)

[Color aditivo (luz, pantallas) 15](#_Toc133332601)

[Color hsl 15](#_Toc133332602)

[Notación hexadecimal y circulo cromático 15](#_Toc133332603)

[Funciones de color sass 15](#_Toc133332604)

¿Qué es Sass?

¿Qué es Sass?

* Preprocesador escrito en Ruby en 2006

http://sass-lang.com/

¿Qué es un preprocesador CSS?

* Es un lenguaje que permite generar CSS a través de una sintaxis similar, pero aprovechando características de la programación.
* Sass - Stylus - Less
* PostCSS no es un preprocesador

Desventajas de css plano

* Los estilos similares deben escribirse manualmente (no se pueden automatizar)
* Las guías de estilo del proyecto no pueden modificarse fácilmente (aunque ya existen variables css)
* No se puede modularizar el código sin afectar el rendimiento del proyecto
* Propenso a sobreescrituras, problemas de especificidad y cascada en proyectos grandes y en equipos de trabajo.

Ventajas de un preprocesador

* Estructurar los estilos en módulos, pero compilarlos a un único archivo CSS resultante
* Programar estilos automáticamente
* Almacenar las guías de estilo del proyecto en variables
* Hacer que proyectos grandes sean mantenibles entre equipos de trabajo.

¿Quién usa Sass?

* Bootstrap

http://getbootstrap.com/

* Foundation

http://foundation.zurb.com/

* Materialize

http://materializecss.com/

Ventajas

* Usado por proyectos líderes
* Gran comunidad y soporte
* Sintaxis sólida y similar a css
* Se puede usar en cualquier entorno (no solo en Ruby) con libsass

Sintaxis de Sass

* Sintaxis Sass (no recomendada, en desuso)
  + Basado en Haml
* Sintaxis Scss (recomendada)
  + Superset de css
  + Todo CSS válido es válido en Sass
* Artículo .sass vs .scss

http://thesassway.com/editorial/sass-vs-scss-which-syntax-is-better

* Comentarios

// Comentarios invisibles en la compilación

/\*Comentarios visibles en la compilación\*/

Compilar Sass

* Probar Sass desde el navegador
  + Sassmeister

https://www.sassmeister.com/

* + Codepen
* Usar un compilador gráfico
  + Prepros

https://prepros.io/

* + Codekit

https://incident57.com/codekit/

* + Koala

http://koala-app.com/

* + Scout

https://mhs.github.io/scout-app/

* Compilar por la línea de comandos
  + ruby sass
    - Instalar Ruby

https://www.ruby-lang.org/en/

Para Windows: http://rubyinstaller.org/

He bajado la 2.3.0 para x64

* + - gem install sass
    - Comandos
      * sass styles.scss styles.css
      * crea el map por defecto
      * sass --watch
      * flags
  + node-sass
    - Instalar Node
    - **npm install -g node-sass**
    - Probar que está instalado **node-sass --version**
    - Comandos
      * De un archivo a otro

node-sass entrada.scss salida.css

* + - * Watch en un directorio de un archivo a otro

node-sass --watch styles.scss styles.css

* + - * Watch de un directorio a otro (se usa punto para directorio actual)

node-sass --watch entrada --output salida

* + - * Flags
* Otras herramientas para compilar
  + Gulp
  + Plugins Brackets
  + Plugins Sass
  + Plugins Atom
  + Plugins Webstorm
* CSS to Sass
  + http://css2sass.herokuapp.com/

Instalar plugin en visual studio code para SASS

Para compilar el CSS se tiene que ir a configuraciones – extensiones – “Live Sass Compiler”

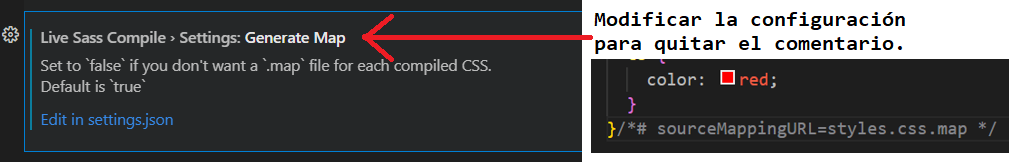
Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

Ingresaras a la ventana de setting

Texto

Descripción generada automáticamente



Variables

Almacena un valor para que puedas usar. En SASS las variables empiezan con $

Texto

Descripción generada automáticamente

Características de las variables:

* Se puede realizar operaciones (+, -, \*, /) aunque no compartan las mismas operaciones.
* Las variables pueden tener un alcance (globales y locales).
* Las variables pueden reescribirse solo para un alcance especifico.
* Las variables SASS se puede colocar en cualquier parte para compilar mediante la interpolación. Se usa el #.

Anidamiento

En el anidamiento por lo general se usa el símbolo “&” y se usa para tener un código más ordenado y anidado. Los casos de uso son:

* Al crear nuevas clases, con pseudo-elementos y pseudo-clases usando el símbolo “&”
* Para las ventanas responsive mediaquery y las animaciones keyframes, sin usar el símbolo “&”

Texto

Descripción generada automáticamente

Tipo de datos:

* String
  + type-of($value: \*) -> devuelve el tipo de dato
  + quote($string: \*) -> coloca comillas dobles
  + unquote($string: \*) -> quita comillas
* Números
  + Los números pueden terne unidad (5px, 2rem, 5s), también se puede realizar operación.
* Colores
  + Tienen funciones especiales para ayudar
* Booleanos
  + Es true o false
* Lists
  + La estructura de los valores de una lista se separa por coma o espacio.
* Maps
  + La estructura de un map es propiedad valor tipo JavaScript.

Nota:

$string: -> indica una cadena

$value: -> una variable o un valor

Texto

Descripción generada automáticamente

@import

Los @import ayudan a estructura el código en distintas partes para poder organizarlo mejor y luego poder llamarlo.

Se puede estructurar un proyecto con muchos archivos sass y carpetas, pero al momento de compilar solo se debería compilar en un solo archivo.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

@extend

Con el @extend puede añadir bloques de código de una clase o agrupados en un placeholder.

Los placeholder comienzan con un símbolo de “%” y se usa para guardar bloques de estilos que luego se pueda reutilizar.

Tener en cuenta que se crea selectores agrupados.

Usar el @extend con una clase, pero ten cuidado con selectores hijos y la especificidad.

Texto

Descripción generada automáticamente

Usar el @extend con un placeholder y me evito problemas de especificidad.

Texto

Descripción generada automáticamente

Condicionales

Texto

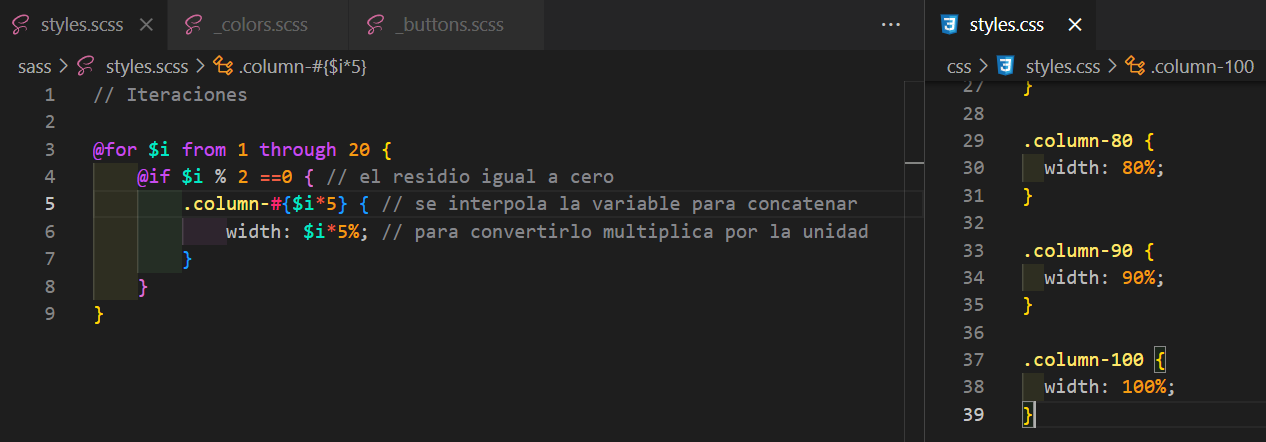
Descripción generada automáticamente

Condicional para controlar las importaciones.

Captura de pantalla de computadora

Descripción generada automáticamente

Ciclo @for



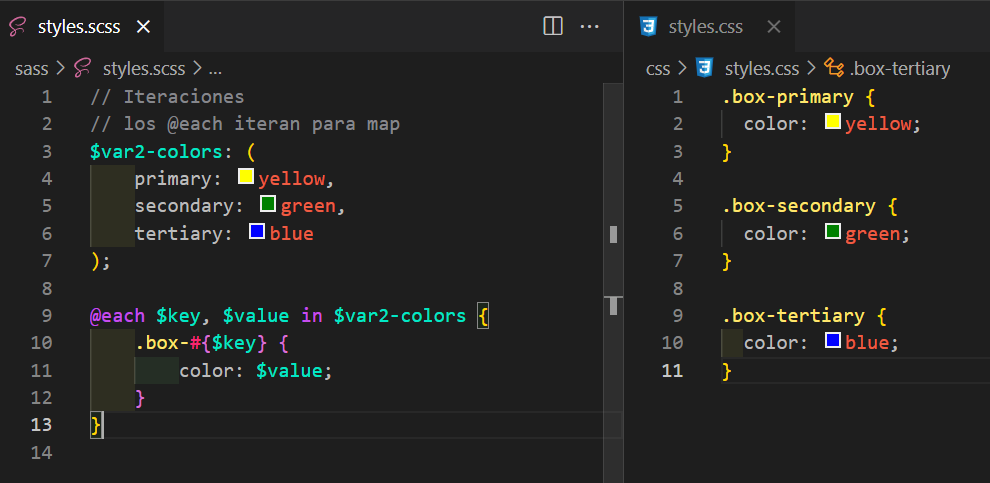
Ciclo @each

El @each se utiliza para recorrer iteraciones de listas.

Captura de pantalla de computadora

Descripción generada automáticamente

El @each se utiliza para recorrer iteraciones de map.



@mixin

Definición de mixins

* Los mixins son bloques de códigos reutilizables que pueden contener selectores, reglas CSS, lógica interna y demás cosas.
* Para que el proyecto sea más organizado, es recomendable crear un módulo para los mixins dentro de la carpeta SCSS.
* Los mixins no necesariamente se tienen que usar dentro de selectores. Cada caso dependerá del contenido que esté dentro del mixin, ya que, dentro del contenido existe la posibilidad de incluir selectores.
* Para usar el mixin se declara la instrucción “@mixin” y se llama con “@include”;

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Con condicionales podemos modularizar el código con la intención de usar o no el código.



Parámetros y argumentos

* Los parámetros son variables que funcionan dentro del mixin, pero no tienen valor definido hasta que nosotros llamamos al mixin.
* Si el mixin tiene parámetros entonces es obligatorio colocarle un valor valido al momento de llamarlo.
* Se llama parámetro cuando se declara en el mixin, pero cuando le das un valor real se llama argumento. Tener en cuenta que el argumento es el valor real y parámetro es la variable que se declara cuando definimos el mixin.

Pantalla de un video juego

Descripción generada automáticamente

Los mixin pueden tener argumentos por defecto y si no le colocas al momento de llamarlo estos toma en valor por defecto.

Texto

Descripción generada automáticamente

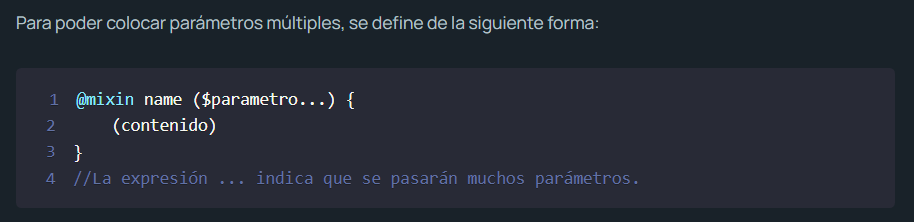
Diferenciar entre parámetros y argumento, la diferencia es que el parámetro es un valor indicado un indicativo de que va a recibir algo, y el argumento es un valor real, es parecido a una función.

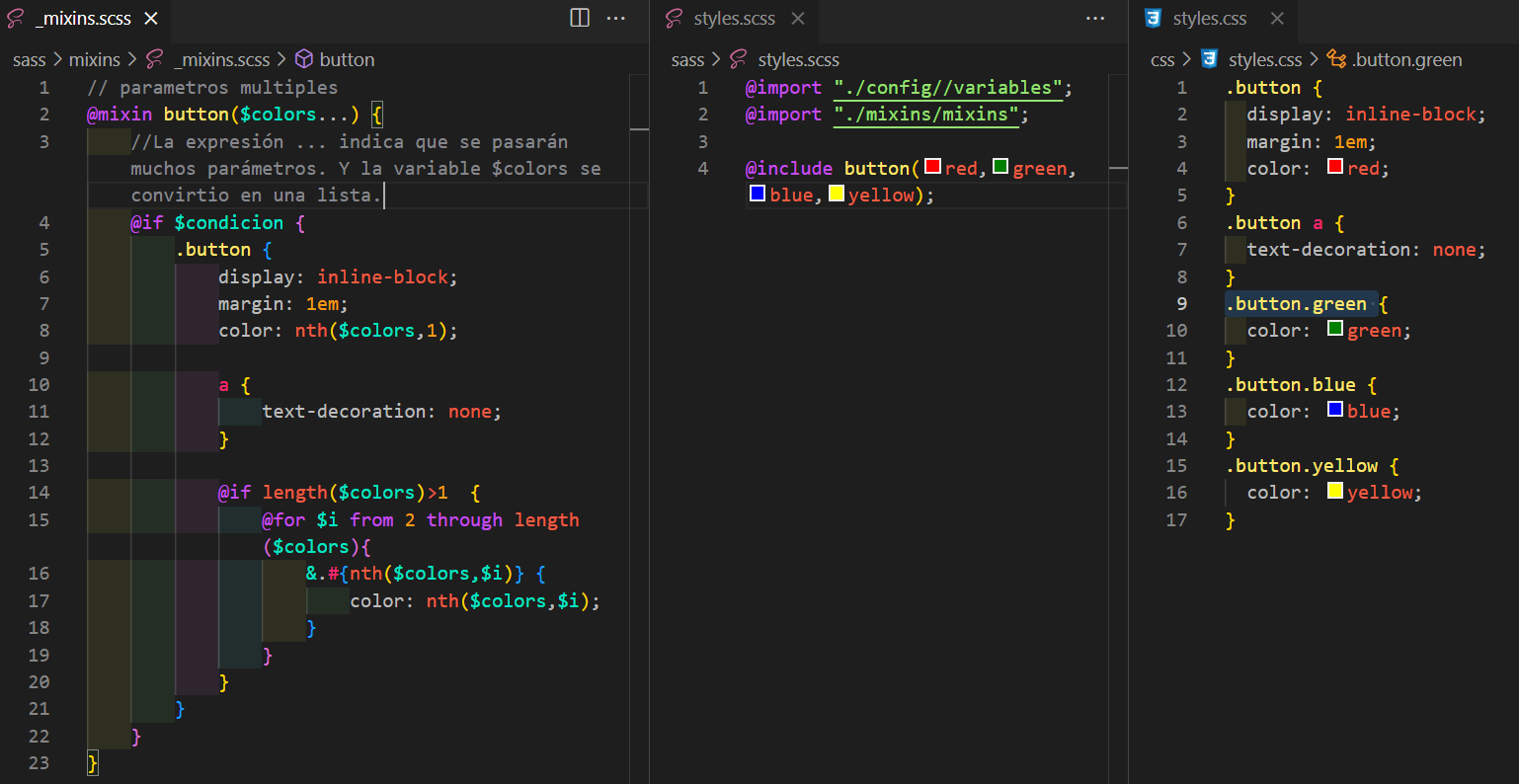
Para resaltar los usos de los mixins con parámetros por defecto o valores predeterminados tales como: (es que tengan un valor)

* Sirve para no depender desde el HTML
* Cuando te dan un HTML desde el Backend.
* Cuando se maneja un CMS y el CMS entrega un marcado HTML estricto.
* Cuando te piden modificar el CSS, pero no puedes acceder al HTML.
* Podemos pasar la estructura como parámetros, es decir acceder a las clases que tiene mi HTML ya entregado y nosotros solo que agregar el CSS.
* No estoy atado a estructuras estrictas y marcado HTML y sus clases, como por ejemplo Bootstrap que para hacer un menú ya cuentan con ciertos div, y clases para armar el menú, los Mixins te liberan de esa dependencia

Parámetros múltiples

* La expresión “…” indica que se pasaran muchos parámetros.
* Y el parámetro se considera una lista.
* Para conseguir el valor de la lista se utiliza “**nth($lista,key)**”





Aquí usamos una variable mapa y las funciones:

* **map-get($map,$key)** = función que obtiene los valores de un mapa.
* **map-key($map)** = obtiene todos los key de un mapa.

Texto

Descripción generada automáticamente

Directiva content

@contect se utiliza para añadir estilos adicionales llamando a un mixin.

Es muy apropiado usarlo en media queries.

Captura de pantalla de computadora

Descripción generada automáticamente

Funciones

Las funciones son trozos de código cuyo objetivo es devolver un valor, es decir, realizan cálculos y devuelven un valor final.

Texto

Descripción generada automáticamente

Las funciones pueden tener parámetros.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Las funciones no son exclusivas de SASS, en CSS también existen funciones tales como url(), calc(), translate(), entre otros. La diferencia radica que en SASS se pueden crear funciones propias, mientras que en CSS no.

Normalmente se tiene la costumbre de trabajar con px, pero por motivos de buenas prácticas es recomendable trabajar con la unidad em.

La em se basa en un tamaño base que es 16px. Ese tamaño puede cambiar de acuerdo con la configuración del navegador, usuario u otras cosas.

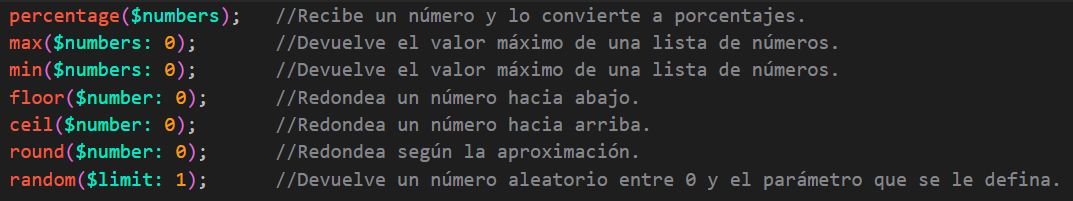
Los mixins pueden devolver todo un bloque de código, selectores, reglas CSS y muchas cosas más. Las funciones devuelven un solo valor.

Funciones de strings

Texto

Descripción generada automáticamente

Funciones de números



Funciones de listas y mapas

Texto

Descripción generada automáticamente con confianza media

Funciones de introspección

Introspección es analizarse a uno mismo, es decir, Sass cuenta con funciones para analizarse a sí mismo algún valor, propiedad, características y demás cosas.

Imagen que contiene Texto

Descripción generada automáticamente

Color

Color sustractivo (pinturas, imprentas)

La luz se reduce, se sustrae. Cuando aumentes mas colores la resultante es casi casi un negro

En el color sustractivo existen 3 colores bases:

* Cyan
* Magenta
* Yellow
* Black

De ahí el nombre CMYK

Color aditivo (luz, pantallas)

En el color aditivo existen 3 colores básicos:

* Red
* Green
* Blue

De ahí proviene el nombre RGB

Las variaciones posibles para cada uno de estos colores es 256, dado esto, los parámetros quedan de la siguiente manera:

RGB (256, 256, 256) = RGB (2^8, 2^8, 2^8)

Tener en cuenta que es desde 0 hasta 255.

Las combinaciones posibles que se pueden tener entre estos 3 es de 16.777.216 de colores, esto es equivalente a la profundidad de colores de 24 bits.

El RGB es la notación que entienden las computadoras, sin embargo, la lectura no es tan amigable con los seres humanos.

Color hsl

Ya que el RGB es una notación complicada de entender para los seres humanos, existe un modo llamado HSL (Hue, Saturation, Lighten).

El HSL es un modo de color muy fácil de entender:

* Hue: es el tono de color (de 0 a 360 grados)
* Saturation: es la intensidad del color, de gris al color puro.
* Lighten: es la cantidad de luz (0% da negro y 100% da blanco, el color puro está en 50%).

En la sintaxis de Sass, no es necesario colocar el signo de porcentaje (%) en los parámetros, al momento de compilar se le agrega el signo automáticamente.

Adobe Photoshop y todos los productos de adobe tienen el modo de color HSB, que no es lo mismo que HSL. EL modo HSB no es compatible con la web.

Affinity designer si trae el modo de color HSL

Para saber el color en el parámetro Hue del HSL, es necesario memorizar los siguientes rangos:

* 0 y 360 → Red
* 60 → Yellow
* 120 → Green
* 180 → Cyan
* 240 → Blue
* 300 → Magenta

<https://color.adobe.com/es/create/color-wheel>

Notación hexadecimal y circulo cromático

La notación Hexadecimal sigue siendo como en RGB, pero con la pequeña diferencia de que se escribe de la siguiente manera: #RRGGBB

Para cada dígito se toma un rango de 0 a 16 con las siguientes notaciones en hexadecimal:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Funciones de color sass

Sass brinda las siguientes funciones para los colores:

Texto

Descripción generada automáticamente

Debugging

Opciones de compilación

Cuando un comando de terminal empieza con 2 guiones -- es probable que dicho comando tenga una versión resumida de usarlo.

La versión corta de --watch es -w

La versión corta de --output es -o

Para que el archivo se compile con los saltos de línea se ejecuta:

node-sass scss -o css --output-style=expanded

En el comando -output-style se tienen las siguientes opciones:

* expanded -> lo expande lo más posible, es mas legible para comprender el css.
* nested -> indentado, por defecto.
* compact -> cada selector ocupa una línea.
* compressed -> código minificado

El comando --source-comments añade comentarios, y en esos comentarios está la especificación de que línea del Sass original ha originado ese CSS.

Sourcemaps

Los Sourcemaps son mapas de origen cuya función es vincular el resultado compilado con el código que le dio origen. Esto se utiliza en todos los lenguajes que hayan sido compilados.

Su principal utilidad radica en que permite trabajar mejor en rastrear funciones y demás cosas.

Para habilitarlo se utiliza el comando \*\*\*\*--source-map de la siguiente manera:

node-sass scss -o css --source-map=true

Debug warn error

En Sass todo lo que comience con @ se le llama directiva.

* @debug es una propiedad que tiene como función inspeccionar un valor y mandar el dato a la consola sin necesidad de ensuciar el CSS.
* @warn tiene como función mandar un mensaje de alerta en la terminal.
* @error lanza un error con un mensaje y detiene el funcionamiento del programa hasta que se solucione dicho problema.

Estructura de proyecto sass

SMACSS

Base: Son todos los estilos aplicados a etiquetas HTML.

Módulos: Son componentes reutilizables. Ejemplo: botones, tarjetas, menús.

Layout: Es la geometría y la posición.

Theme: Tipografías y colores, implica la identidad visual de una marca.

Estado: Elementos que cambian, se animan, etc.

ITCSS

Settings

Tools

Generic

Elements

Objects

Components

Utilities

<https://www.xfive.co/blog/itcss-scalable-maintainable-css-architecture/>

7in1

(<https://gist.github.com/rveitch/84cea9650092119527bc>)

Ejemplo 1: - Estructura repetitiva

$colors: hsl(0,100,50);

@for $i from 1 through 360 {

    $colors: append($colors, $val: hsl($i,100,50), $separator: comma);

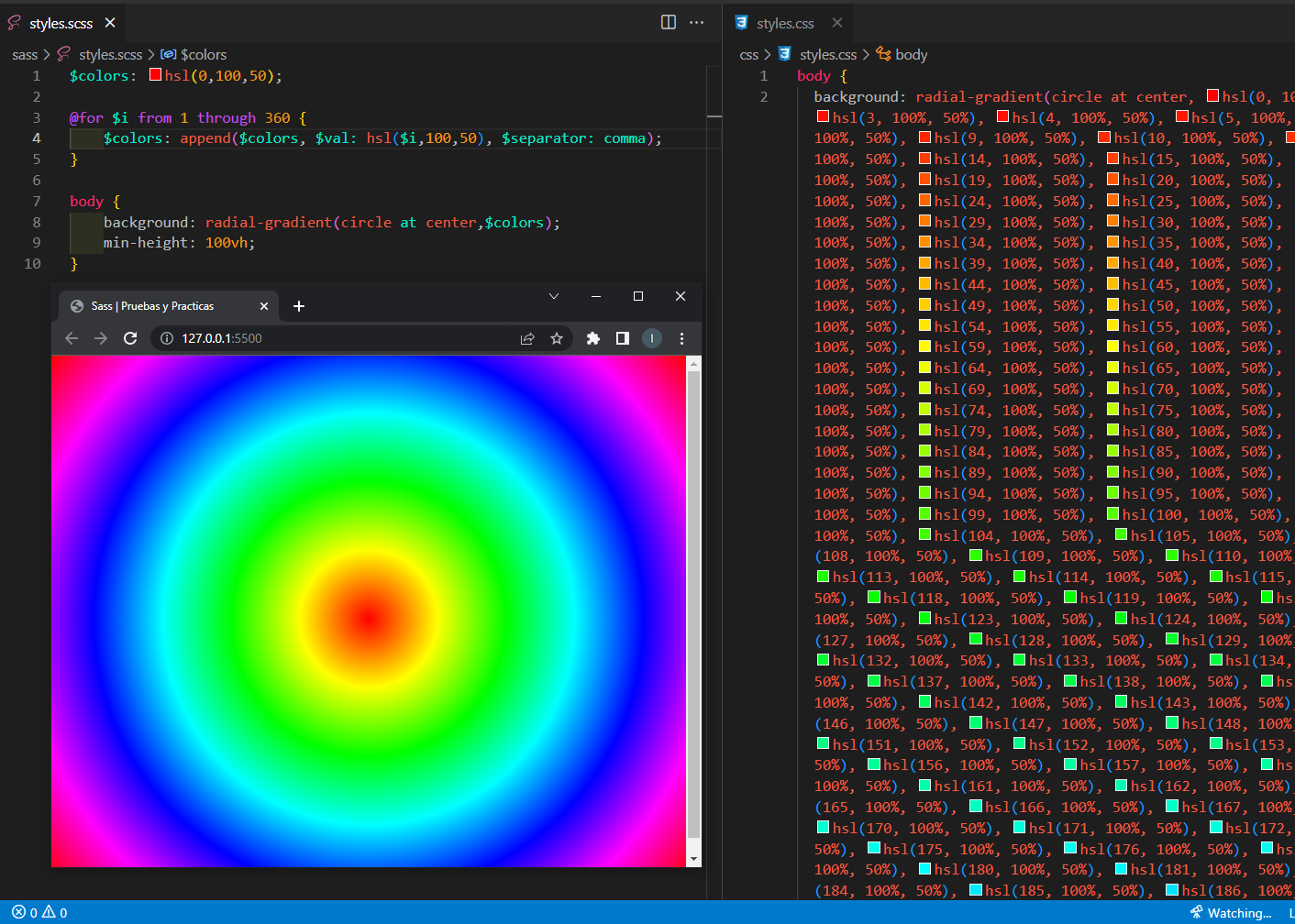
}

body {

    background: radial-gradient(circle at center,$colors);

    min-height: 100vh;

}



Ejemplo 2: - Mixin para responsive web design

Texto

Descripción generada automáticamente

Utilizamos mixin para media queries

Texto

Descripción generada automáticamente

Ejercicio 3

Modulo 4 - ejemplo

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente