Para la instalacion se necesita Nodejs

Npm install -g sass

Para poder compilar el scss a css se pone en consola HAY QUE HACERLO EN LA CONSOLA DE WINDOWS, EN LA CARPETA DEL PROYECTO

sass --watch estilos.scss styles.css

sass –watch(para que este atento a los cambios) estilos.scss (archivo sass) styles.css (el destino)

Ahí se crean los archivos y vas modificando el scss

**Variables:**

$variableNombre: #ccc;

**módulos:**

Los módulos se crean con un \_ al principio x ej: \_encabezado.scss

Ahí les pones los estilos que quieras a cada uno y los podes como “importar” al scss principal:

En el principal pones:

@use ‘base’; (siendo base otro modulo (\_base.scss))

@use ‘encabezado’;  
Entonces ahí se aplican los cambios en el css hechos en los módulos

**Mixins:**

Son funciones. Podes crear como si fuera una función con varias instrucciones y eso después lo aplicas a algún selector con @include

@mixin crearBordes(){

    border:5px solid red;

    border-radius: 10px;

    box-shadow: 0px 0px 5px black;

}

.caja{

    @include crearBordes();

}

.listado{

    @include crearBordes();

}

Podes pasarles parámetros:

@mixin crearBordes($color, $size, $shadow){

    border:$size solid $color;

    border-radius: $size;

    box-shadow: 0px 0px $size $shadow;

}

.caja{

    @include crearBordes(red,10px,black);

}

.listado{

    @include crearBordes(green, 5px, black);

}

**Herencia:**

Se pueden heredar estilos básicos de algo y le podes agregar otros si queres:

%alert-normal{

    border: 1px solid gray;

    background: black;

    color:white;

    padding: 10px;

    width:80%;

    margin:20px auto;

}

.alert{

    @extend %alert-normal;

}

.alert-danger{

    @extend %alert-normal;

    color:red;

    border-color: red;

}

.alert-succes{

    @extend %alert-normal;

    color:green;

    border-color: green;

}

**Operadores:**

Permite Sass hacer operaciones matemáticas.

font-size: 20px + 5px -1px;

Aca saca un porcentaje:

    width: 500px / 1920px \*100%;

Es para no poner la medida a ojo del porcentaje, dividis el tamaño de las cards de dentro de un contenedor y se divide por el tamaño del contenedor. En este caso 500px es la medida de cada una de las cajas y 1920px el tamaño de la caja que las contiene.

**Condicionales:**

h1{

    @if $color-fondo == #ccc {

        color: black

    } @else{

        color: lightcyan

    }

}

    @if $color-fondo == #ccc {

        color: black

    } @else if $color-fondo == white{

        color: lightcyan

    } @else{

        color:pink;

    }

**For:**

No dan dinamicidad a la pagina porque después se compila en un css estático. Pero puede servir para darle a una lista diferentes colores x ej.

$contador: 1;

@for $contador from 1 through 10{

    .listado ul li:nth-child(#{$contador}){

        font-size: $contador + 10px;

    }

}

**PREPROCESADOR LESS:**

En la consola de Windows hacer:

Npm install -g les

Para compilar en la consola de Windows vamos a la carpeta donde esta el proyecto.

lessc estilos.less styles.css

**Variables:**

@bgcolor: lightblue;

@fuente: arial;

@b: body;

@{b}{

    background: @bgcolor;

}

h1{

    font-family: @fuente;

}

Acá es lo mismo que con SASS lo único que se definen con @ podes definir una etiqueta como variable y después llamarla así. También se puede hacer con SASS.

TENES QUE IR COMPILANDO DESPUES DE HACER CADA CAMBIO, UN VIAJE.

Anidacion:

.informacion{

    ul{

        a{

            color:white;

            font-size: 18px;

        }

    }

}

Mixins en les:

.sombras(){

    box-shadow: 0px 0px 10px black;

    text-shadow: 0px 0px 5px black;

}

.informacion{

    .sombras();

}

Con parámetros:

.sombras(@size){

    box-shadow: 0px 0px @size black;

    text-shadow: 0px 0px @size black;

}

.informacion{

    .sombras(5px);

}

//operadores

h1{

    font-size: 30px + 7px;

}

//mixins condicionales, en este caso el mixing se ejecuta si el parametro es true.

.textoNegro(@value) when (@value=true){

    color:black;

}

h1{

    color: blue;

    .textoNegro(false);

}

Bucles:

.loop(@cont) when (@cont > 0){

    .loop((@cont - 1));

    height:(900px @cont);

}

.informacion{

    height: 900px;

    .loop(5);

}

**FLEXBOX:**