**Sass ou SCSS**

**(Syntactically Awesome Style Sheets.)**

* *Não é interpretado pelo navegador.*

**Extensão para VS Code:**

* Live Sass Compiler
* Criamos um arquivo **index.scss** e automaticamente é gerado o arquivo **index.css** traduzindo o Sass para CSS convencional.

**obs.:** também é gerado o arquivo index.css.map. Um arquivo de localização dos elementos css. Podemos descartar esse arquivo.

**Ou Instalando de forma global**:

npm install -g sass

Extensões:

Sass -> oficial

**UTILIZANDO A SINTAXE DO SCSS:**

// Variaveis:

**$primary**: blue;

**$secondary**: gray;

**$warning**: yellow;

**$light**: white;

**$dark**: black;

// Nesting: permite agrupar propriedades.

header {

background: $primary;

padding: 10px;

h1 {

text-align: center;

color: $dark;

}

p {

text-align: left;

color: $light;

}

button {

padding: 15px;

background: $secondary;

}

**&:**hover { // o & adiciona o pseudo elemento no elemento

background: $dark;

}

**&::**after {

content: ‘Um texto aqui’;

}

}

// Import: permite criar arquivos separados para vários elementos do css.

\_header.scss

\_footer.css

.

.

.

etc

// e importá-los dentro de outros arquivos css.

@import ‘./header’; obs.: não precisa colocar a extensão.

@import ‘./footer’;

// mixin: permite criar ‘funções’.

@mixin flexConfig ($dir, $g ){

display: flex;

flex-direction: $dir;

justify-content: center;

align-items: center;

gap: $g;

}

// e aplicá-las em componentes:

header {

@include flexConfig(row, 10px );

}

// extend: permite estender as propriedades de outro elemento.

ex.: criando o arquivo \_contato.scss:

@import ‘header’;

.contato {

@extends ‘header’;

}

**UTILIZANDO A SINTAXE DO SAAS:**

* Não é necessário a utilização de chaves nem ponto e virgula.
* Utiliza-se a indentação.

Utilizamos a extensão .sass para os arquivos de folhas de estilos.