Estudando o SASS

Instalação: (Site de instruções de instalação SASS <https://sass-lang.com/install> )

Acessar o site do SASS: A instalação do SASS é via NPM, portanto o pré requisito para a instação do SASS é a instalação do node : <https://nodejs.org/en/download/>

Após isso, realizaremos o comando npm install -g sass no terminal de comando dentro do diretório do projeto.

Podemos também, rodar o SASS através da extensão live sass compiler do vscode



Pressionado ctrl + shift + p, irá abrir um campo de e vamos procurar por live sass: watch sass



O Sass permite dois tipos de sintaxe a .scss e a .sass a diferença é que a .scss é a mais próxima do css puro é a que utilizaremos nesse aprendizado.

Para auxiliar, instalar a extensão scss formatter no vscode.

Declaração de variáveis no Sass.

As variáveis no sass são declaradas com $ na frente do nome da variável, ex.: $cor-escura: #000

As variáveis só poderão ser acessadas dentro o escopo em que elas foram criadas.

Nesse exemplo, a variável $cor-neutra criada dentro do escopo do seletor .container, não irá funcionar na estilização do seleto body, pois essa variável foi criada em escopo diferente.

Organização e importação de arquivos no sass

Quando criarmos um arquivo com o \_(underline) na frente do nome, estamos dizemos para o sass não compilar esse aquivo.

Para importar esse aquivo em outro arquivo .sass, basta que utilizemos o @import 'nome do arquivo', como mostrado nas imagens a seguir.



No exemplo da imagem acima, o arquivo \_base.scss (chamado de partials), não será compilado.

Arqui estamos importando o arquivo \_base.scss dentro do arquivo main.scss

Encadeamento

O Sass nos permite fazer um encadeamento de estilização, onde dentro de um escopo de estilização de um seletor, podemos encadear a estilização de outro seletor filho. Como mostrado na imagem a seguir:

Aqui temos uma estrutura html.

Aqui vemos a utilização do encadeamento do SASS, onde dentro do escopo de estilização da nossa div de classe container, inserimos a estilização do elemento filho h1 de forma encadeada.

Mixins (Agrupamentos de declarações que poderão ser reutilizadas ).

funciona como se fosse uma função, onde criamos uma estilização padrão que poderá ser chamada dentro de qualquer elemento em que desejamos que tenha essa estilização padrão.

Nesse exemplo, criamos um mixin chamado boxShadow que estiliza um box-shadow padrão e recebe a cor da sombra por parametro.

Chamamos esse mixin utilizando o @include nomeDoMixin, como podemos ver na estilização dos seletores .container e .container2. Ambos irão receber um box-shadow com a cor que passaram por parâmetro.

Condicionais

O Sass também nos permite criar estilização condicional, que funciona como nas maiorias da linguagens de programação utilizando o @if, @else e @else if, como iremos mostrar na imagem abaixo:

Aqui temos uma estilização de cor de texto condicional, onde o nosso mixin receber um valor por parâmetro e verifica se esse valor for igual a danger ele estiliza a cor do texto de vermelho, o valor do parâmetro for alert, a cor do texto será amarelo e diferente disso será preto.

A observação a ser feita aqui é que o texto (srting) a ser comparado não fica entre aspas.

Repetições

@for

Existem duas maneira de utilizar o laço de repetição for no sass

Aqui a primeira forma de utilizar o for com o sass.

Esse exemplo o laço for irá percorrer do número 1 até o número 5.

Observe também que para utilizar uma varável concatenado com uma string, temos que colcar ela dentro de .nome-da-classe#{variavel}, por exemplo.

O outro tipo de for, é usando o to no lugar de through.

A diferença é que o to percorre até um número a menos do que o número indicado, já o through pega até o numero indicado.

Aqui nesse exemplo o laço for irá percorrer do número 1 até o número 4, pois estamos utilizando o from 1 **to** 5.

Lista e laço de repetição @each

No sass podemos criar uma lista chave: valor da seguinte maneira:

$variavel: (chave1:valor1, chave2:valor2, chave3:valor3)

Esse caso é o indicado caso precisemos da utilização da chave para criação de classes, por exemplo, de forma dinâmica.



Uma outra forma de criar uma lista é : $variavel: valor1, valor2, valor3.

Esse é uma forma em que não será necessário a utilização de chave.

Aqui estamos utilizando a lista sem a chave, somente com o valor.

Podemos fazer o laço de repetição @each sem a pré definição de uma variável com a lista, pois podemos utilizar a lista diretamente nele, como mostraremos na imagem a seguir:

Aqui vemos a lista diretamente no laço @each

Herança

O Sass nos permite criar uma herança de estilização de elemento com o uso do

@extend .nome-da-clase, por exemplo.

Aqui, estamos utilizando a herança extendendo a estilização da classe .flex dentro da estilização do elemento body, com isso, o elemento body irá herdar toda a estilização que estiver sendo feita dentro da classe .flex, do qual ele está extendendo, mais a sua estilização particular.

Referência

Podemos criar uma referência a um elemento com o uso do & como mostraremos:

Nesse exemplo o pseudo elemento :hover está sendo referenciado ao elemento p.

Compactando o arquivo.

Parei aqui: https://youtu.be/BaI8dHUthLA?t=2592