SUMÁRIO

1. O QUE É SASS

É comum vermos os termos LESS, Foundation e SASS, quando inseridos no meio do desenvolvimento web. Esses nomes foram dados à pré-processadores e frameworks de folhas de estilo para auxiliar na produtividade de códigos, principalmente no que diz respeito a repetição de uma mesma ação, diversas vezes.

Sabemos que CSS é uma linguagem de estilização não programável, pois é uma linguagem declarativa em cascata. Não existem tarefas em CSS, nem processos programados, nem laços de repetição, ou sequer variáveis, são apenas atributos que devem ser declarados de forma a moldar os elementos estruturados em HTML.

Destas limitações, surgem as soluções dos pré-processadores para CSS, que permitem que façamos a estilização à moda das linguagens de programação. Desta forma, ao invés de, por exemplo, copiarmos uma mesma classe diversas vezes para aplicarmos diferentes estilizações, podemos criar um laço de repetição e um bloco de códigos único que incluirá todos as possíveis e necessárias aplicações da classe – coisa que, em CSS comum, seria realizada em diversas linhas, e em blocos diferentes.

1.1. EXTENSÃO SASS E SCSS

Embora estejamos discorrendo sobre o pré-processador SASS, dando a entender que os arquivos possuem esta exata extensão, ela não é mais a padrão quando nos referimos a codificação em SASS.

Isto se deve ao fato de que, quando utilizada a extensão antes padrão, SASS, a codificação se torna distinta daquela realizada em CSS, não existindo uma compatibilidade entre as duas. Isto é, SASS é incompatível com a sintaxe de CSS, enquanto que SCSS é compatível, podendo ambas serem utilizadas no mesmo arquivo.

Esta possibilidade facilita o aprendizado, pois não demanda a assimilação de uma sintaxe completamente diferente daquela de CSS. Em SASS, por exemplo, a codificação se dá por indentações, não existindo chaves para delimitar blocos de códigos e escopos. Já com a extensão do arquivo em SCSS, a sintaxe usual de CSS se mantém, e é combinada com as novas features do pré-processador. Portanto, e além de ser o novo padrão, a utilização SCSS é sempre preferível.

1.2. COMO UTILIZAR O PRÉ-PROCESSADOR SASS?

O arquivo de estilização pré-processado não é utilizado diretamente no documento HTML, sendo, na verdade, convertido para uma folha de estilo CSS. Ou seja, de qualquer forma, o arquivo final utilizado para estilização é o padrão CSS. A utilização de um pré-processador está para os ganhos de produtividade, isto é, de realização de uma folha de estilo mais eficientemente. Assim, quando começar a mexer no Sass, ele pegará seu arquivo Sass pré-processado e o salvará como um arquivo CSS normal que você pode usar em seu site.

A maneira mais direta de fazer isso acontecer é em seu terminal. Assim que o Sass estiver instalado, você pode compilar seu Sass para CSS usando o sass comando. Por exemplo, se existir um arquivo.scss que queira converter para CSS, poderá digitar, no terminal, no caso do Windows, ‘sass arquivo.scss arquivo.css’.

Outra possibilidade, é a de utilizar o comando ‘watch’, para que, em tempo real, o arquivo seja convertido para CSS. Isto é, mesmo enquanto codificado, a cada modificação e código escrito corretamente no arquivo.scss, automaticamente ocorrerá a reescrita no arquivo.css. Para isto, escrevemos no terminal ‘sass –watch arquivo.scss:arquivo.css’.

Outras duas formas de utilizar, é (i) com aplicativos de interface gráfica, como o Koala, que permite o gerenciamento dos arquivos.scss, e (ii) no VS Code, com a utilização de plugins, como o Hero Compiler, que realiza a mesma tarefa do comando de terminal ‘watch’ supracitado.

2. FEATURES DO SASS

2.1. VARIÁVEIS

Pense nas variáveis como uma forma de armazenar informações que você deseja reutilizar em sua folha de estilo. Você pode armazenar coisas como cores, pilhas de fontes ou qualquer valor CSS que achar que deseja reutilizar. Sass usa o $ símbolo para tornar algo uma variável. Veja abaixo, a codificação e a conversão para CSS:

SCSS Sass

```scss
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

CSS

```css
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

Figura 1 Fonte: https://sass-lang.com/guide

2.2. NESTING

O Sass permitirá que você aninhe seus seletores CSS DE uma maneira que siga a mesma hierarquia visual de seu HTML.

SCSS Sass

```scss
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

CSS

```css
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

Figura 2 Fonte: https://sass-lang.com/guide

2.3. MIXINS

Algumas coisas em CSS são um pouco tediosas de escrever, especialmente com CSS3 e os muitos prefixos de fornecedores que existem. Um mixin permite que você faça grupos de declarações CSS que deseja reutilizar em todo o seu site. Você pode até mesmo passar valores para tornar seu mixin mais flexível.

SCSS Sass

```scss
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}
.box { @include transform(rotate(30deg)); }
```

CSS

```css
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}
```

Figura 3 Fonte: https://sass-lang.com/guide

2.4. HERANÇA

Este é um dos recursos mais úteis do Sass. Usar @extendpermite compartilhar um conjunto de propriedades CSS de um seletor para outro. Para isso, uma classe de espaço é reservada, sendo um tipo especial de classe que só imprime quando é estendida e pode ajudar a manter seu CSS compilado limpo e organizado.

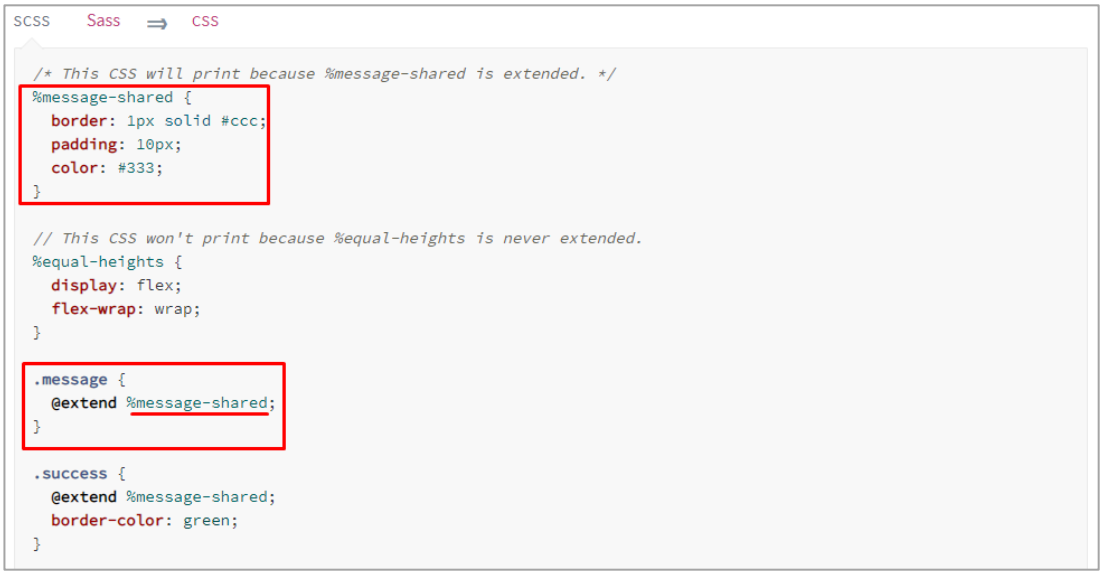
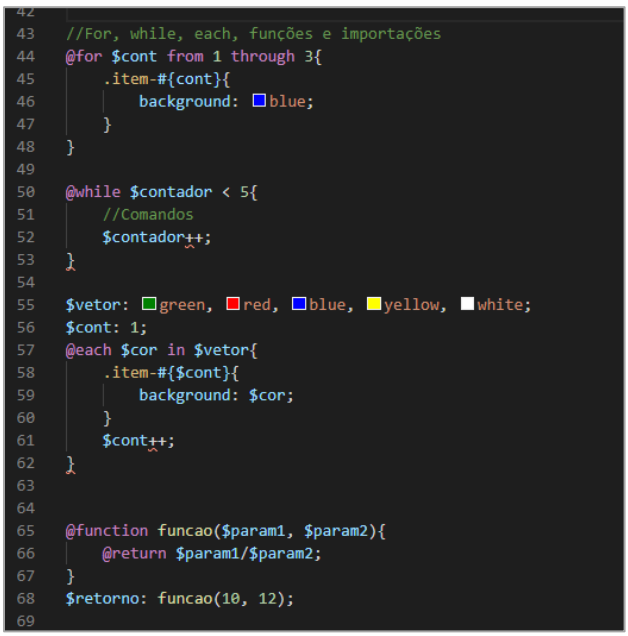


Figura 4 Fonte: https://sass-lang.com/guide

2.5. ARRAYS DE ATRIBUTOS, LAÇOS DE REPETIÇÃO E FUNÇÕES

Algumas outras possibilidades incluem estas estruturas bem conhecidas, para aqui operar dinamicamente com diversos atributos:



3. FONTES

https://sass-lang.com/guide

https://tableless.com.br/sass-um-outro-metodo-de-escrever-css/