

Como criar variáveis com o Sass

Transcrição

O Sass é um pré-processador CSS, um CSS com superpoderes, como ele mesmo se denomina. Um desses superpoderes é exatamente o uso de variáveis.

Para chamarmos o Sass no CSS utilizamos o `$` :

```
$cor-padrao: #c24e4b;
```

E dentro do CSS, em todos os lugares que essa cor aparece, fazemos como no exemplo:

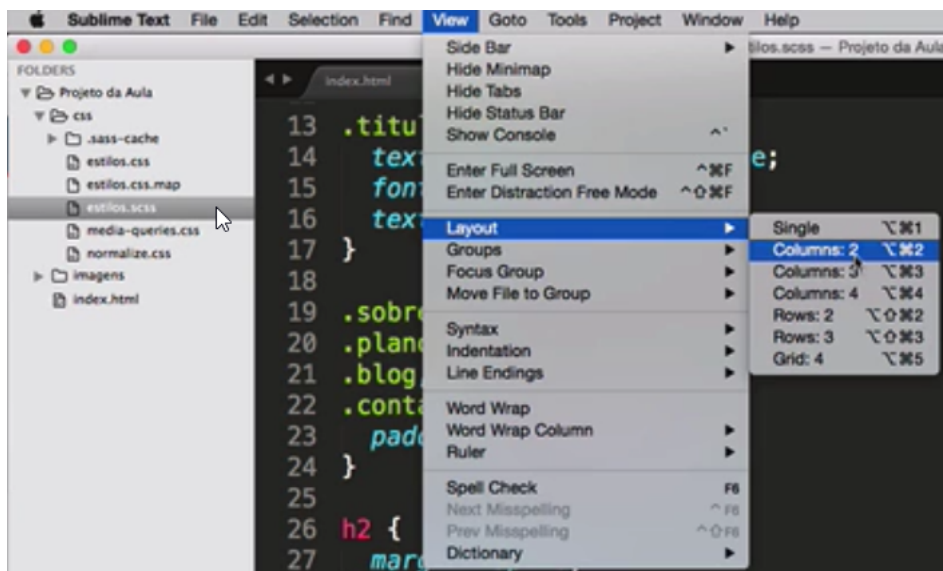
```
/** Header **/  
header {  
  border-top: 5px solid $cor-padrao;  
  ...  
}
```

Para que essas mudanças aconteçam, precisamos transformar o arquivo em SCSS, uma das linguagens que o Sass utiliza, a mais parecida com o CSS padrão. Então renomeamos o arquivo com a extensão ".scss".

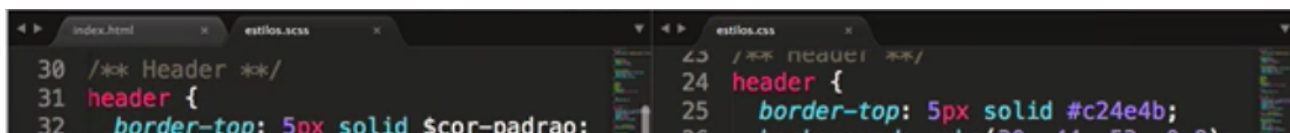
O Sass é um pré-processador CSS, então a primeira coisa que devemos fazer é pedir para que ele compile o arquivo "estilos.scss", e faremos isso por meio do terminal. O primeiro passo é entrar na pasta do projeto (em que está o arquivo a ser compilado), depois compilamos o arquivo .scss para criar um .css . O comando da compilação é:

```
sass estilos.scss:estilos.css
```

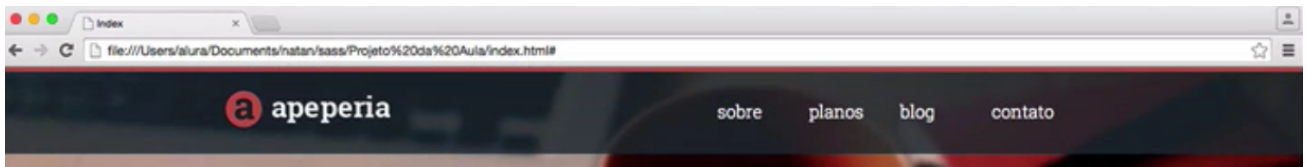
Atente para o nome dos arquivos, que devem estar iguais aos que estão na pasta do projeto. Embora o terminal não apresente nada específico ao usarmos esse comando, ao pedir para que ele liste os arquivos da pasta, vemos que agora existe o arquivo estilos.css . Abrindo o nosso código, percebemos que o arquivo .scss mantém-se igual, mas o .css que foi criado está com todas as cores trocadas como queríamos! Também vale notar variável que colocamos no início do arquivo .scss já não existe no arquivo .css . Para facilitar a comparação entre os dois arquivos, podemos dividir o nosso editor em duas colunas, usando o seguinte caminho:



Que resulta nessa visualização:

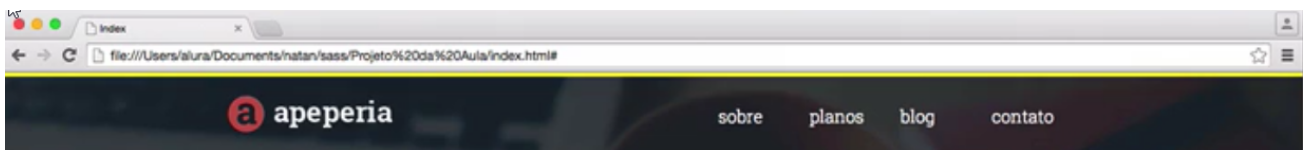


Em resumo, ao compilarmos, o Sass passa todas as informações de um arquivo para o outro automaticamente. Então poderemos escrever com facilidade no '.scss' que o Sass faz sozinho o trabalho passar para o '.css'. Ao testarmos no site, vemos que funcionou e a cor passou do azul inicial para o vermelho que o cliente escolheu:



E se quisermos trocar a cor padrão novamente? Voltando ao arquivo `.scss`, podemos alterar a variável `$cor-padrao` para amarelo (`yellow`). Salvando o arquivo e atualizando o site, nada muda. Ainda está o mesmo vermelho. Voltando ao arquivo `.css`, o código da cor também não mudou.

Isso acontece porque o arquivo foi compilado só uma vez, logo depois que mudamos do azul. Para que uma segunda mudança aconteça, precisamos voltar ao terminal e pedir que o Sass compile novamente. Para isso, usamos o mesmo comando de compilação. Voltando ao arquivo `.css`, o código de cor estará atualizado. Testando o site no navegador, vemos que ele também está com a cor certa.



Vemos que inserir uma variável, facilita muito uma mudança de cor. Mas ainda é necessário compilar no terminal a cada uma delas. Não seria mais prático se pudéssemos automatizar a compilação? O Sass nos permite isso, com o comando `--watch` (do inglês, "assistir" ou "observar"). Com os arquivos que estamos trabalhando nesse projeto, o código no terminal fica assim:

```
sass --watch estilos.scss:estilos.css
```

E o terminal nos retorna da seguinte maneira:

```
sass --watch estilos.scss:estilos.css
>>> Sass is watching for changes. Press Ctrl-C to stop.
```

O que se traduz por "O Sass está observando em busca de mudanças. Pressione Ctrl-C para parar". Agora, qualquer mudança feita na variável `$cor-padrao`, passará automaticamente para o arquivo `.css`. Podemos aproveitar isso para voltar à cor que o cliente pediu, a `#c24e4b` é notável como essa mudança acontece mais rápido. Logo que inserimos o código hexadecimal da cor no arquivo `.scss`, o Sass reconhece e muda no arquivo `.css`. Salvando e testando no site, vemos que já está da cor escolhida, num processo muito mais ágil que o anterior.

Podemos inserir outras variáveis para, por exemplo, alterar outra cor que se repete no site. Escolhemos a cor escura que se repete no *header*, na área onde estão os planos oferecidos e no *footer* e a chamaremos de `cor-auxiliar`. Assim como para a cor padrão, criaremos uma variável no início do arquivo `.scss`:

```
$cor-padrao: #c24e4b;  
$cor-auxiliar: #1e2c35;
```

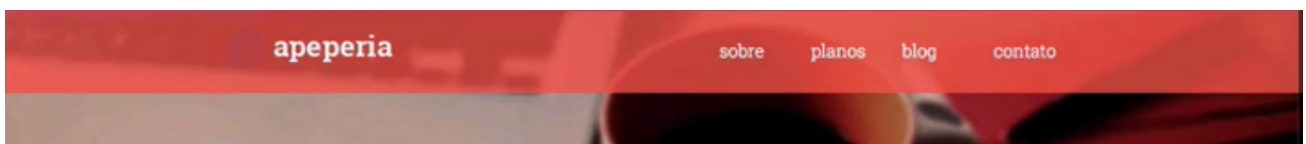
Em seguida, precisamos chamar a variável `cor-auxiliar` em todos os lugares em que essa variável é usada, trocando seu código por `#1e2c35`. No caso desse arquivo, o código da cor do *header* está em RGBA. Como proceder então?

```
header {  
  border-top: 5px solid $cor-padrao;  
  background: rgba(30, 44, 53, 0.8)  
  height: 90px  
  width: 100%  
  position: absolute;  
}
```

Como os três primeiros números do RGBA correspondem à cor RGB, podemos trocá-los pela nossa variável. Ficará assim:

```
background: rgba($cor-auxiliar, 0.8)
```

Assim que salvamos o arquivo, tanto o nosso arquivo `.css` quanto o site estarão prontos para receber qualquer mudança nessa cor. Para testar, colocaremos `tomato` como `cor-auxiliar`, e assim que salvarmos o arquivo, podemos ir para o site e teremos o seguinte resultado:



Como o Sass permite que você crie uma variável para o que for necessário, o que é especialmente vantajoso com valores ou partes de código que se repetem bastante. Por enquanto é só. Até a próxima!

