

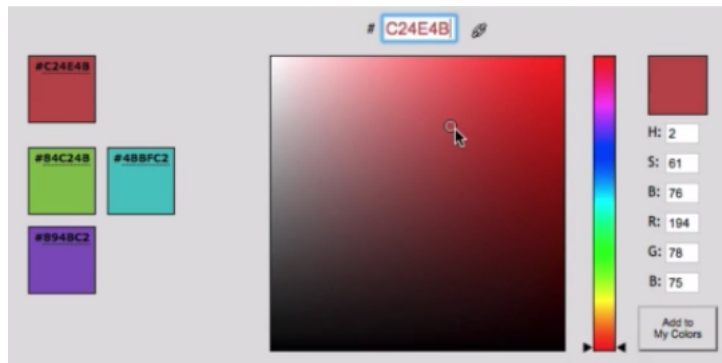
## Alterações nas cores

### Transcrição

Se o cliente pedir para mudar a cor principal do site para uma mais escura, devemos ir à nossa pasta "helpers > variaveis" e verificar o código hexadecimal da cor padrão atual.

```
// Variaveis usadas no meu Sass
$cor-padrao: #c24e4b;
$cor-auxiliar: #1e2c35;
$cor-branca: #fff;
```

Para escolher uma mais escura, podemos usar uma ferramenta online chamada [Color Picker](http://www.colorpicker.com/) (<http://www.colorpicker.com/>).



Copiando o código de \$cor-padrao no campo do Color Picker, podemos mudá-la com o cursor na área de cor. Para deixar mais escura, basta descer o seletor e copiar o código quando ele exibir uma cor que agrada. Como exemplo, escolhemos a cor de código #96403E. Atualizando no código no arquivo de estilos, temos:

```
// Variaveis usadas no meu Sass
$cor-padrao: #96403E;
$cor-auxiliar: #1e2c35;
$cor-branca: #fff;
```

E no navegador, o site está assim:



E caso o cliente não goste e queira voltar para a cor inicial? Pode ser que já não tenhamos como dar "CTRL + Z", se o editor de texto já tiver sido fechado, por exemplo. Então teríamos que procurar o código da cor anterior em algum e-mail ou arquivo antigo. Para evitarmos isso, é melhor usarmos um jeito de trocar a cor que preserve o código original.

O Sass tem funções de cor que poderão nos ajudar com isso. Uma delas se chama `darken` (do inglês, escurecer) e ele pede um argumento, que será o código da cor. Usando-o na `$cor-padrao` :

```
// Variaveis usadas no meu Sass
$cor-padrao: darken(#c24e4b);
$cor-auxiliar: #1e2c35;
$cor-branca: #fff;
```

Salvando esse arquivo, o `.css` dá o seguinte erro, acusando que a função está com o número errado de argumentos:

```
Error: wrong number of arguments (1 for 2) for 'darken'
on line 2 of helpers/variaveis.scss
from line 1 of estilos.scss
```

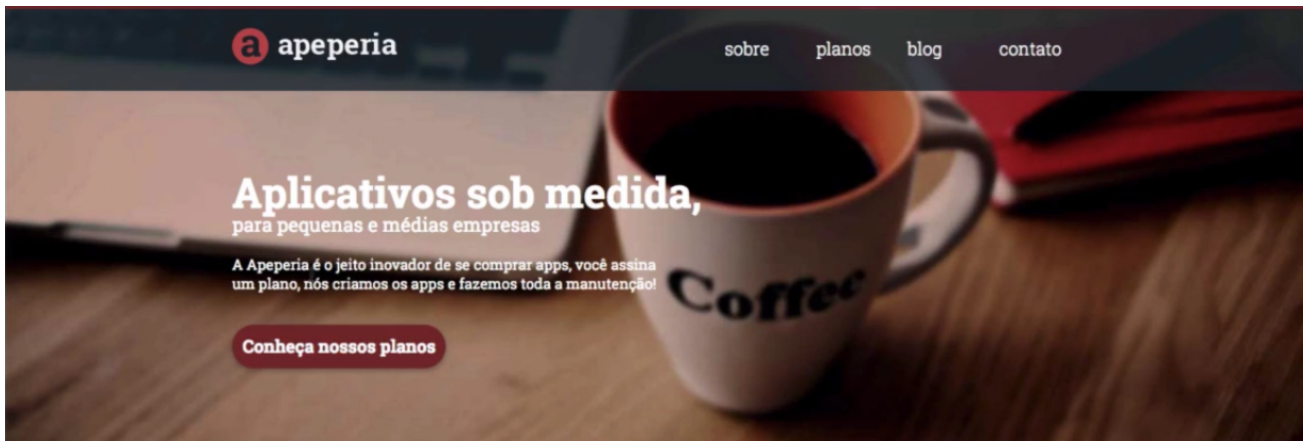
Ele está nos comunicando que essa função não depende só da cor inicial, mas também de quão mais escura você quer deixá-la, em porcentagem.

```
// Variaveis usadas no meu Sass
$cor-padrao: darken(#c24e4b, 20%);
$cor-auxiliar: #1e2c35;
$cor-branca: #fff;
```

Salvando o arquivo, o erro no `.css` já não acontece mais. Quando verificamos como está o código da cor está nesse arquivo, vemos:

```
header {
  border-top: 5px solid #7d2c2a;
  background: rgba(30, 44, 53, 0.8);
  height: 90px;
  width: 100%
  position: absolute;}
```

Identificamos que o código da nova cor é `#7d2c2a` , que é o resultado da compilação da função `darken` . Para ver o resultado no site, vamos ao navegador:



E vemos que, de fato, a cor está mais escura. E se quisermos deixar mais escura ainda, basta aumentar a porcentagem dentro da função `darken`.

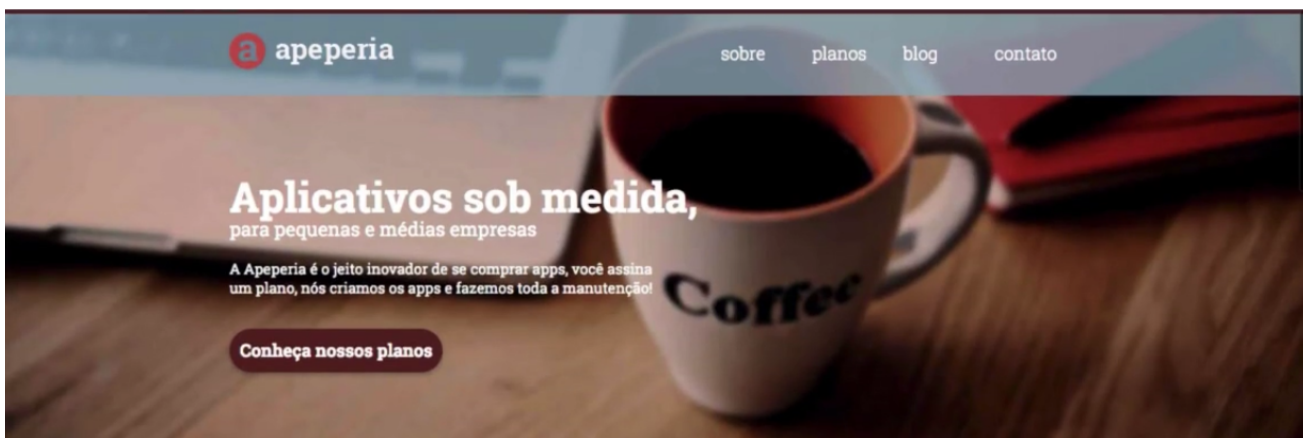
E se quisermos deixar a cor mais clara? Também é possível, basta usar a função `lighten`. Vamos usá-la na `$cor-auxiliar` e aumentar a porcentagem da função `darken`:

```
// Variaveis usadas no meu Sass
$cor-padrao: darken(#c24e4b, 30%);
$cor-auxiliar: lighten(#1e2c35, 50%);
$cor-branca: #fff;
```

Uma dica útil é que o segundo número não precisa ser expresso como uma porcentagem. A função o identifica como tal automaticamente, então podemos deixar os argumentos assim:

```
// Variaveis usadas no meu Sass
$cor-padrao: darken(#c24e4b, 30);
$cor-auxiliar: lighten(#1e2c35, 50);
$cor-branca: #fff;
```

Atualizando o site, vemos que funcionou bem:



Fica claro que as funções de cor facilitam bastante a nossa vida. Até a próxima!