

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="sass/css/funcoes_especiais.css" />
<title>Funções Especiais</title>
</head>
<body>

<!--
```

Funções Especiais

O que são Funções Especiais?

Funções Especiais são funções que estão presentes tanto no Sass quanto no Css. Elas são especiais porque podemos cria-las no Sass e elas serão replicadas no CSS, pois elas estão presentes nos dois sistemas.

Quando temos funções assim, em vez de apresentar no arquivo final CSS somente os resultados da função no seletor que está utilizando, ela vai replicar a função mesmo no CSS, e elas serão carregadas em tempo real no próprio arquivo CSS final.

Detalhe importante!!!

Todas as chamadas de funções especiais retornam strings sem aspas, ou seja, mesmo que contatemos 2 strings com aspas, as 2 strings serão concatenadas e apresentadas com apenas 1 aspa em cada ponta, ou seja, não haverão 4 aspas, mas somente 2, como deve ser para que a string possa ser lida.

OBS: Para mais detalhes veja "Fundamentos do Sass nº24";

-->

```
<p id="p1">Hcode Code</p>
<p id="p2">Hcode Code</p>
```

```
</body>
</html>
```

ARQUIVO SASS...

```
$font: "../fonts/Open_Sans/"; //Usamos uma variável apenas para indicar o caminho

//A regra @font-face é uma função especial do Sass que pode ser ramificada para o CSS final
@font-face {
  font-family: "OpenSans Regular"; //Nome dado a essa fonte
  src: url("#{ $font }OpenSans-Regular.ttf); /*Perceba que contatemos o caminho contido na variável $font, junto com o
                                             nome do arquivo em questão, mas para isso usamos um template string*/
  font-weight: 400;
}

@font-face {
  font-family: "OpenSans Light"; //Nome dado a essa fonte
  src: url($font + "OpenSans-Light.ttf"); /*Perceba que aqui também fizemos uma concatenação, porém, note que o nome
                                             do arquivo em questão está entre áspas, e mesmo assim, ao juntar o
                                             caminho da variável que também está entre áspas, com o nome do arquivo,
                                             a função conseguiu fazer a concatenação correta do caminho.*/
  font-weight: 300;
}

#p1 {
  font-family: "OpenSans Regular";
  font-size: 80px;
}

#p2 {
  font-family: "OpenSans Light";
  font-size: 80px;
}
```

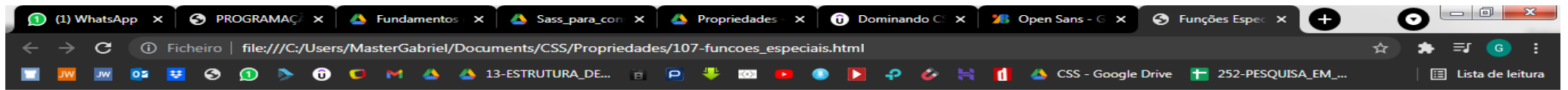
ARQUIVO CSS...

```
@font-face { /*Perceba que a regra está presente tanto no Sass quando no CSS*/
  font-family: "OpenSans Regular";
  src: url(../fonts/Open_Sans/OpenSans-Regular.ttf); /*Veja como fica a concatenação feita com template string*/
  font-weight: 400;
}
@font-face {
  font-family: "OpenSans Light";
  src: url("../fonts/Open_Sans/OpenSans-Light.ttf"); /*Veja como fica a concatenação feita de forma natural por
                                                    meio de função especial*/
  font-weight: 300;
}
#p1 {
  font-family: "OpenSans Regular";
  font-size: 80px;
}

#p2 {
  font-family: "OpenSans Light";
  font-size: 80px;
}

/*# sourceMappingURL=funcoes_especiais.css.map */
```

RESULTADO NO NAVEGADOR...



Hcode Code

Hcode Code

