

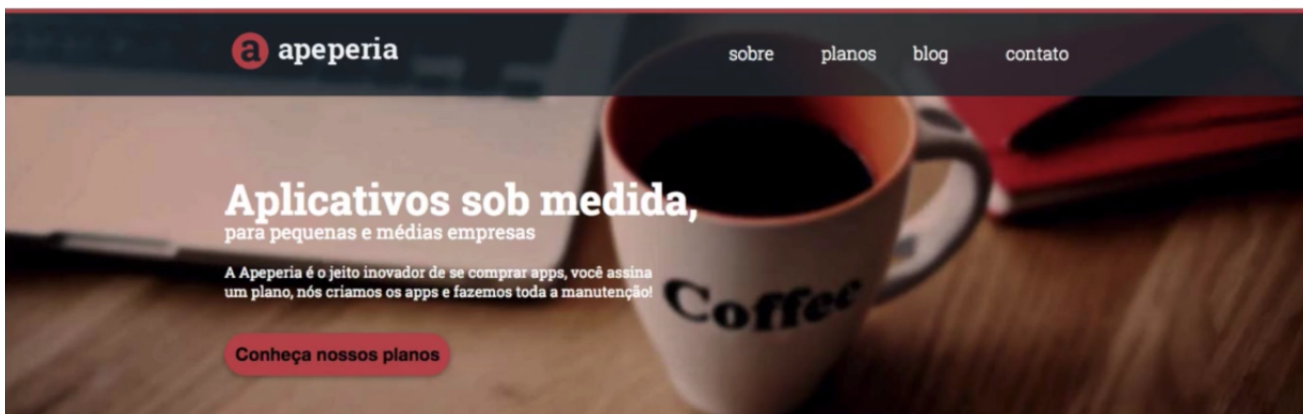
Importando o normalize

Transcrição

Quando olhamos nosso arquivo Sass, sentimos falta da importação do `normalize`, de quem o layout depende muito. Quando abrimos o `index.html`, vemos que ainda está marcado como se o `normalize` estivesse na pasta "css":

```
<head>
  <meta charset="UTF-8">
  <title>Index</title>
  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="css/media-queries.css">
  <link href='http://fonts.googleapis.com/css?family=Roboto+Slab: 400,300,700' rel='stylesheet' />
</head>
```

Mas, na realidade, esse arquivo foi colocado na pasta "base". Quando conferimos no navegador, vemos que o site não está como deveria.



Podemos alterar o caminho direto nesses arquivos.

```
<head>
  <meta charset="UTF-8">
  <title>Index</title>
  <link rel="stylesheet" href="css/base/normalize.css">
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="css/media-queries.css">
  <link href='http://fonts.googleapis.com/css?family=Roboto+Slab: 400,300,700' rel='stylesheet' />
</head>
```

E ao atualizarmos, o site fica com a aparência correta:



Nesse arquivo HTML, nós estamos puxando três arquivos `.css`, além da fonte do Google. Focando no `normalize` e no estilos, podemos pensar em juntá-los em um arquivo só. Essa é outra das boas práticas do CSS que podemos aplicar no Sass. Para fazer essa concatenação, voltamos ao nosso arquivo `.scss` e importamos o `normalize`:

```
@import 'helpers/variaveis';
@import 'helpers/mixins';

@import 'base/normalize';
@import 'layout/geral';
@import 'layout/header';
@import 'layout/destaque';
@import 'layout/sobre';
@import 'layout/planos';
@import 'layout/blog';
@import 'layout/contato';
@import 'layout/footer';
```

Quando o salvamos, o arquivo `.css` imediatamente dá erro. Ele não encontra o arquivo porque, quando não especificamos a extensão, o Sass pensa que ela é `.scss`. Como não é o caso, vamos especificar:

```
@import 'helpers/variaveis';
@import 'helpers/mixins';

@import 'base/normalize.css';
@import 'layout/geral';
@import 'layout/header';
@import 'layout/destaque';
@import 'layout/sobre';
@import 'layout/planos';
@import 'layout/blog';
@import 'layout/contato';
@import 'layout/footer';
```

Agora o arquivo está sendo compilado corretamente e o nosso `.css` está começando assim:

```
@import url(base/normalize.css);
```

Mas não é tão interessante manter o comando de importação nesse arquivo. O melhor é que o código do `normalize` entre diretamente nesse trecho de código, assim como acontece com os outros arquivos importados. E qual é a

diferença entre esses arquivos e o `normalize`? A extensão.

Podemos mudar a extensão desse arquivo o renomeando para `normalize.scss` direto na pasta "base". Agora basta deletar a extensão antiga no `estilos.scss`:

```
@import 'helpers/variaveis';
@import 'helpers/mixins';

@import 'base/normalize';
@import 'layout/geral';
@import 'layout/header';
@import 'layout/destaque';
@import 'layout/sobre';
@import 'layout/planos';
@import 'layout/blog';
@import 'layout/contato';
@import 'layout/footer';
```

Agora, no arquivo `.css` o código do `normalize` precede os de layout como se tivesse sido colado antes dele, que é o que pretendíamos. No `index` já não precisamos da chamada do `normalize`, então a deletaremos:

```
<head>
  <meta charset="UTF-8">
  <title>Index</title>
  <link rel="stylesheet" href="css/estilos.css">
  <link rel="stylesheet" href="css/media-queries.css">
  <link href='http://fonts.googleapis.com/css?family=Roboto+Slab: 400,300,700' rel='stylesheet' />
</head>
```

Pronto, terminamos essa etapa e ficaremos por aqui. Até a próxima!