

02

Apresentação do mixin

Transcrição

Ao darmos mais uma olhada no arquivo `.scss`, identificamos que o valor de `border-radius` de `0.3em`, que diz respeito ao tamanho da fonte, aparece bastante. Quando uma coisa se repete, é vantajoso criar uma variável no Sass. Dessa vez chamaremos essa variável de `$raio`:

```
$raio: 0.3em;
```

Como antes, substituíremos todas as ocorrências do valor `0.3em` pela variável. Para conferir se o Sass já compilou o arquivo com as alterações que fizemos, vamos ao terminal:

```
>>>Change detected to: estilos.scss
```

Essa mensagem mostra que ele detectou as nossas mudanças e o arquivo `.css` já está identificando a variável criada.

Analizando novamente o objeto em que inserimos a nova variável, notaremos um prefixo antes de `border-radius`:

```
.plano button {
  background: $cor-padrão;
  border: 0;
  color: $cor-branca;
  width: 210px;
  height: 38px;
  font-size: 1.2em;
  -webkit-border-radius: $raio;
  border-radius: $raio;
  box-shadow: 0 2px 6.65px 0.35px rgba
  margin-left: 2em;
}
```

Sabemos que esse prefixo `-webkit` se refere ao navegador. Se precisarmos acrescentar para o Mozilla, precisamos acrescentar o prefixo `-moz`. Se quisermos colocar para o Opera também, acrescentamos o prefixo `-o`.

```
.plano button {
  ...
  -webkit-border-radius: $raio;
  -moz-border-radius: $raio;
  -o-border-radius: $raio;
  border-radius: $raio;
  ...
}
```

Como a nossa variável está aplicada em outros lugares, precisaríamos acrescentar o `border-radius` com os novos prefixos neles também. Mas fazer isso manualmente seria trabalhoso e demorado. Esse é um problema da repetição, mas é justamente com ela que o Sass nos ajuda. Nos outros casos, isolamos o valor que se repete e criamos uma

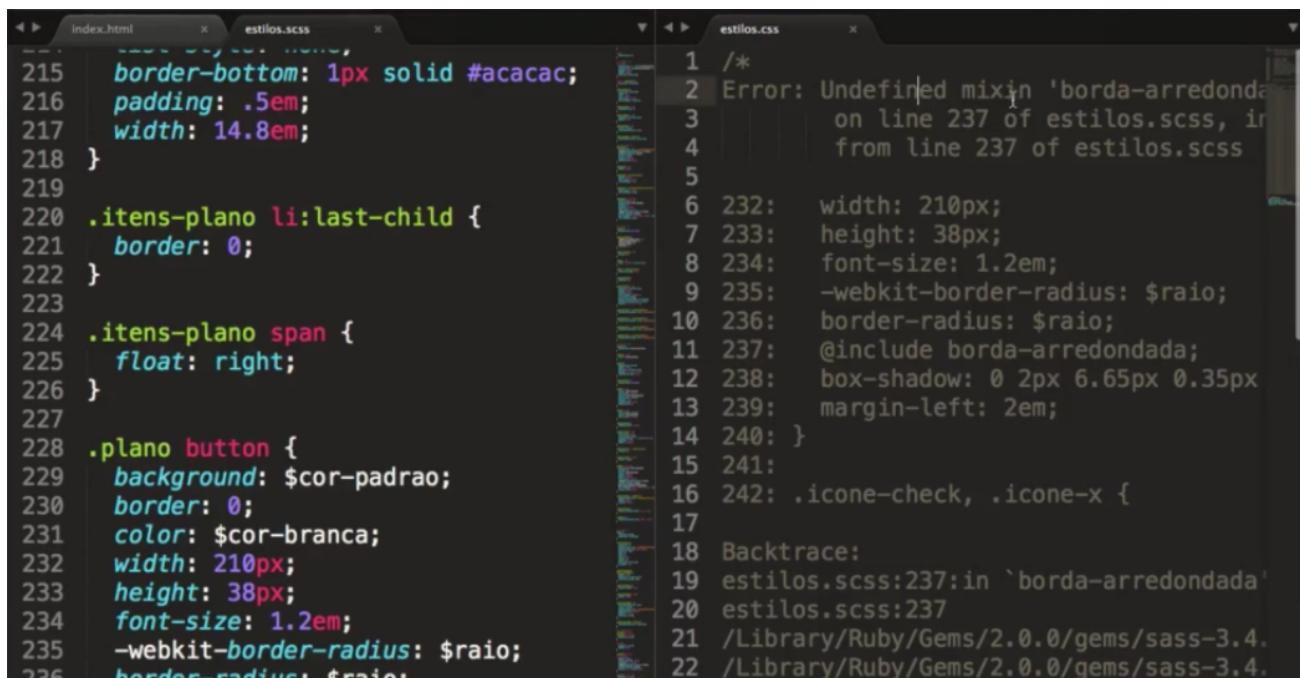
variável. Nesse caso, a solução para o nosso problema é o *mixin*: um trecho de código transformado em variável. A sintaxe é parecida com a do CSS e o *mixin* do `border-radius` se chamará "borda arredondada" e ficará assim:

```
@mixin borda-arredondada {
  -webkit-border-radius: $raio;
  border-radius: $raio;
}
```

Assim como as variáveis, o *mixin* deve ser chamado nos lugares em que precisamos dele. Colocaremos o *mixin* em `plano button` usando o comando `@include`.

```
.plano button {
  ...
  -webkit-border-radius: $raio;
  border-radius: $raio;
  @include borda-arredondada;
  ...
}
```

Entretanto, assim que o arquivo é salvo, temos uma surpresa: o arquivo `.css` apresenta erro!



```
index.html x estilos.scss x estilos.css x
215  border-bottom: 1px solid #acacac;
216  padding: .5em;
217  width: 14.8em;
218 }
219
220 .itens-plano li:last-child {
221   border: 0;
222 }
223
224 .itens-plano span {
225   float: right;
226 }
227
228 .plano button {
229   background: $cor-padrao;
230   border: 0;
231   color: $cor-branca;
232   width: 210px;
233   height: 38px;
234   font-size: 1.2em;
235   -webkit-border-radius: $raio;
236   border-radius: $raio;
237
238   /* Error: Undefined mixin 'borda-arredondada' on line 237 of estilos.scss, imported from line 237 of estilos.scss
239   */
240 }
241
242 .icone-check, .icone-x {
243
244   width: 210px;
245   height: 38px;
246   font-size: 1.2em;
247   border: 1px solid #acacac;
248   border-radius: 50%;
249   background-color: #fff;
250   background-size: 100% 100%;
```

Esse erro aponta que o *mixin* é indefinido. Quando comparamos o *mixin*, no topo do arquivo, com o que inserimos em `plano button`, percebemos que há uma diferença. No primeiro, o nome está como "borda arredondada", e no segundo "borda arredondada". O terminal também aponta esse erro, com a mensagem:

```
>>>Change detected to: estilos.scss
      error estilos.scss (Line 237: Undefined mixin 'borda arredondada'.)
```

Ao corrigirmos o nome do *mixin* para "arredondada", o Sass corrige o erro e o arquivo `.css` é atualizado. Mas olhando o código do `plano button` neste arquivo, vemos uma repetição:

```
.plano button {
  background: #c24e4b;
  border: 0;
  color: #fff;
  width: 210px;
  height: 38px;
  font-size: 1.2em;
  -webkit-border-radius: 0.3em;
  border-radius: 0.3em;
  -webkit-border-radius: 0.3em;
  border-radius: 0.3em;
  box-shadow: 0 2px 6.65px 0.35px rgba
  margin-left: 2em;
}
```

O conteúdo do *mixin* se repete porque deixamos as duas linhas originais e o `@include` do *mixin* no arquivo `.scss`. Para evitar esse problema, basta deletar as duas linhas originais:

```
.plano button {
  ...
  @include borda-arredondada;
  ...
}
```

Com essa questão resolvida, podemos aplicar o *mixin* em todas as ocorrências de `border-radius` usando `@include borda-arredondada`. Acompanhando o arquivo `.css` enquanto incluímos o *mixin*, não há mudanças aparentes – o que não é motivo para alarme, porque o código do *mixin* é igual ao que já estava anteriormente. Para garantir que está tudo funcionando como deveria, podemos testar incluindo uma alteração no *mixin*. Acresentaremos o prefixo para o Mozilla, para o Opera e o `ms`, que é o do Internet Explorer:

```
@mixin borda-arredondada {
  -webkit-border-radius: $raio;
  -moz-border-radius: $raio;
  -o-border-radius: $raio;
  -ms-border-radius: $raio;
  border-radius: $raio;
}
```

Ao salvar o arquivo `.scss` e conferir o `.css`, vemos que em todos os pontos em que usamos o *mixin*, o código está atualizado com o que inserimos.

Novamente, o Sass nos dá a opção de mudar em apenas um lugar (o *mixin*), o que precisaríamos mudar em vários. Continuaremos usando o *mixin* nas próximas aulas. Até lá!