

Vídeo 3

Transcrição

Agora o designer me enviou um email, pedindo que a borda do botão de destaque fosse mais arredondada. No entanto, ele quer que os demais botões permaneçam como estão.

Para isto, criaremos um novo mixin no `estilos.less` que chamaremos de `.borda-mais-arredondada` e ficará posicionado logo abaixo do `.borda-arredondada`.

```
.borda-mais-arredondada {  
  -webkit-border-radius: @raio-da-borda;  
  -o-border-radius: @raio-da-borda;  
  border-radius: @raio-da-borda;  
}
```

No entanto, se queremos deixar a borda mais arredondada, teremos que aumentar o valor do raio. Logo, teremos que criar uma nova variável.

```
@cor-padrao: #c24e4b;  
@cor-auxiliar: #1e2c35;  
@cor-branca: #000;  
@raio-da-borda: 0.3em;  
@raio-da-borda-mais-arredondada: 2em;
```

Agora, onde estava o `@raio-da-borda`, substituiremos por `@raio-da-borda-mais-arredondada`:

```
.borda-mais-arredondada {  
  -webkit-border-radius: @raio-da-borda-mais-arredondada;  
  -o-border-radius: @raio-da-borda-mais-arredondada;  
  border-radius: @raio-da-borda-mais-arredondada;  
}
```

Faremos a mesma substituição no `.destaque button`:

```
.destaque button {  
  margin-top: 1em;  
  background: #66DDDD;  
  border: 0;  
  padding: .6em;  
  font-size: 1.2em;  
  .borda-mais-arredondada;  
  .sombra-padrao;  
  font-weight: bold;  
}
```

Após salvarmos as alterações, poderemos ver como o botão ficou no navegador:



Está tudo funcionando corretamente, mas ficar chamando com frequência o `borda-arredondada` e depois, o `borda-mais-arredondada` não é o ideal, porque estamos repetindo código. Antes de mostrar como poderíamos fazer de outra forma, vou apagar a atrocidade de acabei de criar: `.borda-mais-arredondada` e o `@raio-da-borda-mais-arredondada`. **Evitem** adicionar algo assim no seu código.

O que realmente queremos e fazer o seguinte:

```
.destaque button {
  margin-top: 1em;
  background: #66DDDD;
  border: 0;
  padding: .6em;
  font-size: 1.2em;
  .borda-arredondada, amigo, e o valor vai ser 2em;
  .sombra-padrao;
  font-weight: bold;
}
```

Queremos que ao chamarmos o `.borda-arredondada` o LESS saiba que o valor será `2em`. O que fizemos acima é uma **brincadeira**, a forma correta de passar um valor direto na chamada do mixin é adicionando o valor entre parentêses:

```
.destaque button {
  margin-top: 1em;
  background: #66DDDD;
  border: 0;
  padding: .6em;
  font-size: 1.2em;
  .borda-arredondada(2em);
  .sombra-padrao;
  font-weight: bold;
}
```

Mas se tentarmos fazer um teste, o Terminal irá reclamar. Veremos que ele retornará uma mensagem de erro:

```
>> File "css/estilos.less" changed.
Running "less:development" (less) task
>> RuntimeError: No matching definition was found for `.borda-arredondada(
2em)` in css/estilos.less on line 102, column 3:
>> 101   font-size: 1.2em;
>> 102   .borda-arredondada(2em);
>> 103   .sombra-padrao;
Warning: Error compiling css/estilos.less Use --force to continue.
```

Tivemos problema no momento da compilação. Isto aconteceu porque quando criamos o mixin `.borda-arredondada` não definimos que sempre que ele for chamado, deverá ser passado um valor. A solução é adicionar `(@raio-da-borda)` :

```
.borda-arredondada(@raio-da-borda) {
  -webkit-border-radius: @raio-da-borda;
  -o-border-radius: @raio-da-borda;
  border-radius: @raio-da-borda;
}
```

Agora, sempre que usarmos o `.borda-arredonda` teremos que passar um valor, no nosso caso, `2em` . Nos outros botões, queremos que o valor seja `0.3em` :

```
.plano button {
  background: #66DDDD;
  border: 0;
  color: #fff;
  width: 210px;
  height: 38px;
  font-size: 1.2em;
  .borda-arredondada (0.3em);
  .sombra-padrao;
  margin-left: 2em;
}
```

Faremos o mesmo em `.contato input` e `.contato button` :

```
.contato input,
textarea {
  width: 100%;
  margin-bottom: .8em;
  padding: 4px 8px;
  .borda-arredondada(0.3em);
  border: 2px solid #ccc;
}

.contato button {
  margin: 0 auto;
  background: #66DDDD;
  border: 0;
  color: #fff;
  width: 210px;
  height: 38px;
  font-size: 1.2em;
  .borda-arredondada(0.3em);
}
```

```
.sombra-padrao;  
}
```

Agora, sim! Ao salvarmos as alterações no arquivo LESS, veremos que o CSS receberá as modificações. No navegador, o botão de destaque estará com as bordas bem arredondadas:



Porém, da maneira como montamos o código, se precisarmos mudar o valor do `borda-arredondada` **normal** - ou seja, o que tem o valor de `0.3em` - teremos que fazer mudanças em várias partes. Uma solução seria criarmos uma variável, por exemplo, `@raio-padrao` :

```
@cor-padrao: #c24e4b;  
@cor-auxiliar: #1e2c35;  
@cor-branca: #fff;  
@raio-da-borda: 0.3em;  
@raio-padrao: 0.3em;
```

Porém, neste caso, nós sobrescreveríamos o valor do botão de destaque `2em` que adicionamos anteriormente.

```
.destaque button {  
  margin-top: 1em;  
  background: #66DDDD;  
  border: 0;  
  padding: .6em;  
  font-size: 1.2em;  
  .borda-arredondada(2em);  
  .sombra-padrao;  
  font-weight: bold;  
}
```

Por isso, não iremos mais passar usar as variáveis `@raio-da-borda` e `@raio-padrao` . Será melhor passar um valor padrão, e caso o desenvolvedor queira, terá a opção de definir o valor. Mas caso ele não queira especificar, será usado o valor padrão `0.3em` .

```
.borda-arredondada(@raio-da-borda: 0.3em) {  
  -webkit-border-radius: @raio-da-borda;  
  -o-border-radius: @raio-da-borda;  
  border-radius: @raio-da-borda;  
}
```

É como se criássemos uma variável diretamente no mixin. No `.destaque button` manteremos o `2em` passado entre parênteses, mas nos itens em que adicionamos o `0.3em`, iremos remover os valores. Um deles é o `.plano button` que ficará assim:

```
.plano button {  
  background: #66DDDD;  
  border: 0;  
  color: #fff;  
  width: 210px;  
  height: 38px;  
  font-size: 1.2em;  
  .borda-arredondada ();  
  .sombra-padrao;  
  margin-left: 2em;  
}
```

Veja que o parênteses de `.borda-arredondada` ficou vazio, assim ele irá usar o valor padrão.

Quando o LESS for compilado, veremos que o valores corretos surgirão no `estilos.css` :

```
.plano button {  
  background: #66DDDD;  
  border: 0;  
  color: #fff;  
  width: 210px;  
  height: 38px;  
  font-size: 1.2em;  
  -webkit-border-radius: 0.3em;  
  -o-border-radius: 0.3em;  
  border-radius: 0.3em;  
  box-shadow: 0 2px 6.65px 0.35px rgba(0, 0, 0, 0.3);  
  margin-left: 2em;  
}
```

O valor `0.3em` apareceu.

Se visualizarmos a página no navegador, poderemos perceber que os botões estão com as bordas que definimos. É possível confirmar isso acessando o "Inspecionar elementos":



Então, é interessante ao criarmos este mixin definirmos um valor padrão, que será utilizado nas declarações do CSS.