

04

Alterações na borda arredondada

Transcrição

Suponhamos que o designer do site tenha se comunicado e pedido para que a borda do botão da área de destaque (destaque button) fosse mais arredondada do que é, e nos passado o valor de 50px. Por enquanto, o botão está assim:



Como criamos o variável `$raio`, podemos usá-la agora com o valor que o designer nos passou:

```
$raio: 50px;
```

Ao substituir o valor no arquivo `.scss`, o Sass já troca o valor no arquivo `.css`. Ao abrirmos este arquivo para confirmar se a mudança realmente foi feita:

```
.destaque button {  
  ...  
  -webkit-border-radius: 50px;  
  border-radius: 50px;  
}
```

E quando salvamos e atualizamos no site, confirmamos que funcionou:



Mas, rolando o site, vemos que a mudança foi aplicada em lugares que o designer não pediu, como os botões na área dos planos:

<u>Planos</u>	
Plano Básico	Plano Enterprise
R\$ 500,00	R\$ 1500,00
Aplicativos 1 Manutenções Programadas 5 Helpdesk 24hs ✘ App SEO ✓ Atendente Exclusivo ✘ Sistema Operacional 1	Aplicativos 3 Manutenções Programadas 20 Helpdesk 24hs ✓ App SEO ✓ Atendente Exclusivo ✘ Sistema Operacional 3
Comprar	Comprar
Plano Corporation	R\$ 9500,00
Aplicativos 5 Manutenções Programadas ∞ Helpdesk 24hs ✓ App SEO ✓ Atendente Exclusivo ✘ Sistema Operacional 6	Aplicativos 5 Manutenções Programadas ∞ Helpdesk 24hs ✓ App SEO ✓ Atendente Exclusivo ✘ Sistema Operacional 6
Comprar	Comprar

E também no formulário de contato, prejudicando inclusive a leitura de um dos campos:

Contato

Digite seu nome...

Digite seu e-mail...

Digite o assunto...

Mensagem...

Enviar

Apenas o botão da área de destaque deveria ter mudado, os outros deveriam ter ficado com 0.3em. Para resolver esse problema, podemos criar um segundo *mixin* chamado `borda-mais-arredondada` , e codificá-lo o *mixin* da seguinte maneira:

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

A variável `$raio` é a que está sendo usada no outro *mixin* e que precisa voltar para 0.3em. Então criaremos outra variável para esse novo *mixin*, que se chamará `$raio-grande` . As duas variáveis ficarão assim:

```
$raio: 0.3em;
$raio-grande: 50px;
```

Agora que temos a variável e o *mixin*, precisamos associá-las ao `destaque button` com o comando `@include` :

```
.destaque button {
  ...
  font-size: 1.2em;
  @include borda-mais-arredondada;
  @include sombra-padrão;
  font-weight: bold;
}
```

Voltando ao arquivo `.scss`, vemos que o código dos *mixins* está repetitivo.

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

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

Precisamos alterar a variável `$raio` para `$raio-grande` no `maxin borda-mais-arredondada`:

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

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

Ao conferir o arquivo `.css`, constatamos que os valores foram atribuídos corretamente: o raio do `destaque button` está com `50px` e o dos demais continua `0.3em`. No browser também está tudo com a aparência esperada.

Se precisarmos fazer botões com um terceiro tipo de borda e criarmos outro *mixin* para isso, ficará mais repetitivo ainda. Uma solução é usar apenas um *mixin* e especificar o valor de `borda-arredondada` em cada objeto, usando `()`.

Voltando ao `destaque button` :

```
.destaque button {
  ...
  font-size: 1.2em;
  @include borda-arredondada(50px);
  @include sombra-padrão;
  font-weight: bold;
}
```

Ao fazer essa alteração e salvar, o arquivo `.css` imediatamente dá erro. Isso acontece porque no `mixin` borda-arredondada não há nenhum comando que permita especificar valores. Podemos fazer isso inserindo a variável `$raio` no `mixin`.

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

Após salvar essa alteração, o arquivo `.css` continua apresentando erro, e nos mostra a seguinte mensagem:

Error: Mixin borda-arredondada `is` missing on line 239 of estilos.scss, in 'borda-arredondada' fi

Quando conferimos a linha 239, vemos que é a do `insert` do `mixin` borda-arredondada no objeto `plano button`. Agora esse `mixin` precisa de um parâmetro na sua chamada. Para corrigir esse problema, colocaremos o valor `0.3em`:

```
.plano button {
  ...
  font-size: 1.2em;
  @include borda-arredondada(0.3em);
  @include sombra-padrão;
  margin-left: 2em;
}
```

E repetiremos esse procedimento em todos os objetos em que colocamos o `mixin`. Salvando essas alterações, vemos que o arquivo `.css` voltou a funcionar normalmente e os valores de `border-radius` que definimos individualmente na chamada do `mixin` estão aplicados. É interessante poder definir os valores individualmente, mas, nesse caso, estamos repetindo muito o valor `0.3em`. Para que não precisemos mudar em cada lugar na próxima alteração, podemos definir no `mixin` um valor padrão. Para definir esse valor padrão, voltamos ao `mixin` e acrescentamos o valor dentro dos parênteses que inserimos inicialmente:

```
@mixin borda-arredondada($raio: 0.3em) {
  -webkit-border-radius: $raio;
  border-radius: $raio;
}
```

Isso elimina a necessidade da variável `$raio` (que será deletada) e da especificação de `0.3em` em todos os objetos que colocamos o `mixin` com esse valor. Manteremos a especificação de `50px` no `destaque button`. Por exemplo, o `plano button` ficará assim:

```
.plano button {
  ...
  font-size: 1.2em;
  @include borda-arredondada;
  @include sombra-padrão;
  margin-left: 2em;
}
```

Para testar se esse padrão funciona, vamos definir como o novo valor padrão no `mixin` 10px. Ao salvar o arquivo, os valores correspondentes no arquivo `.css` estão corretos: 10px. Mas o `destaque button`, está com o valor correto? Conferindo no arquivo `.css`, vemos que ele continua com o valor de 50px. Verificando no navegador, vemos que o botão na área de destaque está com a aparência correta:



E os demais estão mais arredondados que inicialmente, como o esperado.



Portanto, se precisamos de um valor padrão, podemos defini-lo direto na criação do `mixin` e ainda assim manter esse valor modificável quando necessário. Até a próxima!