

03

Media query como variável

Transcrição

Revendo o código do *media query*, vemos que ele depende de uma largura máxima (`max-width`). Como exemplo, temos a inserção de `@media` no `container`

```
.container {
  width: 940px;
  margin: 0 auto;

  @media (max-width: $ponto-de-quebra) {
    .container
      width: 90%
  }
}
```

Se quisermos que ele dependa de uma largura mínima, precisaremos mudar uma a uma todas as ocorrências do `@media`. Depois de aprendermos tantas maneiras de facilitar com o Sass mudanças de código que se repetem, esse método não parece muito funcional.

Já isolamos o valor do `$ponto-de-quebra` em uma variável, e podemos agora isolar a própria `@media` em uma variável. Supondo que seja um *media query* para *mobile*, o arquivo de variáveis fica assim:

```
//Variaveis usadas no meu Sass
$cor-padroao: darken(#c24e4b, 0);
$cor-auxiliar: lighten(#1e2c35, 0);
$cor-branca: #fff;

$ponto-de-quebra: 980px;

$mq-mobile: (max-width: 980px);
```

Criada a variável, podemos aplicá-la:

```
.container {
  width: 940px;
  margin: 0 auto;

  @media $mq-mobile {
    .container
      width: 90%
  }
}
```

Ao salvar, o arquivo compilado apresenta um erro. Ele não está identificando o *media query*, mesmo que tenhamos colocado o texto certo na variável. Mas, quando usamos o recurso de variável no Sass, ele assume que estamos

inserindo um valor. Para concatenar o texto do *media query*, precisamos usar uma sintaxe diferente, com uma cerquilha (#) e chaves:

```
.container {
  width: 940px;
  margin: 0 auto;

  @media #{$mq-mobile} {
    .container
      width: 90%
  }
}
```

O erro ainda persiste, mas dessa vez o arquivo compilado nos diz que não é um valor CSS válido. Agora que declaramos na inserção da variável que ela é um texto, voltamos ao arquivo de variáveis. Nele, também precisamos mostrar que essa é uma variável do tipo *string*, usando aspas duplas:

```
//Variaveis usadas no meu Sass
$cor-padrao: darken(#c24e4b, 0);
$cor-auxiliar: lighten(#1e2c35, 0);
$cor-branca: #fff;

$ponto-de-quebra: 980px;

$mq-mobile: "(max-width: 980px)";
```

Depois de salvar essa alteração, o erro some e o trecho de código no qual aplicamos a variável está assim no arquivo compilado:

```
.container {
  width: 940px;
  margin: 0 auto;
}

@media (max-width: 980px) {
  .container .container {
    width: 90%; }
}
```

Percebemos que há repetição de `.container`. Revendo a entrada no arquivo `.scss`, fica claro o porquê: estamos repetindo o seletor. Corrigindo, o código fica assim:

```
.container {
  width: 940px;
  margin: 0 auto;

  @media #{$mq-mobile} {
    width: 90%
  }
}
```

Agora o código no arquivo compilado fica correto:

```
.container {
  width: 940px;
  margin: 0 auto;
}

@media (max-width: 980px) {
  .container {
    width: 90%; } }
```

Precisamos aplicar a variável que criamos nas outras ocorrências de `@media`, no arquivo `header.scss`:

```
header {
  border-top: 5px solid $cor-padrao;
  background: rgba($cor-auxiliar, 0.8);
  height: 90px;
  width: 100%;
  position: absolute;

  @media #{$mq-mobile} {
    height: auto;

    h1 {
      max-width: 50%;
      margin: 0 auto;

      img {
        max-width: 100%;
        margin: .5em auto;
        display: block;
      } // fim do img
    } // fim do h1
  } // fim do media querie
} // fim do header

header .container {
  position: relative;

  @media #{$mq-mobile} {
    position: static;
  }
}
```

Se quisermos mudar de `max-width` para `min-width`, basta voltar ao arquivo de variáveis e fazer essa alteração nele. Se a variável estiver sendo chamada da maneira correta, concatenada pela cerquilha e pelas chaves, o Sass já fará as trocas sozinho. Isolar o *media query* inteira é uma ótima alternativa a isolar apenas o seu valor. Até a próxima!