

Vídeo 2

Transcrição

Uma outra melhoria que podemos fazer é salvar o valor `16` que agora se repete no `footer .container` :

```
footer .container {  
  padding-top: 16 * 3px;  
  height: 16 * 6.5px;  
  position: relative;  
}
```

Para facilitar, podemos adicionar a variável `@tamanho-da-fonte-do-html` no `variaveis.less` :

```
// Variaveis  
@cor-padrao: #c24e4b;  
@cor-auxiliar: #1e2c35;  
@cor-branca: #fff;  
  
@mobile: ~(max-width: 980px);  
  
@tamanho-da-fonte-do-html: 16;
```

Iremos substituir o valor `16` do `footer.less` pela variável recém criada:

```
footer .container {  
  padding-top: @tamanho-da-fonte-do-html * 3px;  
  height: @tamanho-da-fonte-do-html * 6.5px;  
  position: relative;  
}
```

Para fazer um teste, vamos substituir a unidade de medida de `px` para `mm` no `height` . A conversão ocorreria corretamente no CSS:

```
footer .container {  
  padding-top: @tamanho-da-fonte-do-html * 3px;  
  height: 104mm;  
  position: relative;  
}
```

Milímetros(mm) é uma unidade que pode ser trabalhada no CSS, sendo mais utilizada em trabalhos impressos.

Voltaremos o nosso código para a unidade de medida `px` :

```
footer .container {  
  padding-top: @tamanho-da-fonte-do-html * 3px;  
  height: 104px;  
}
```

```
position: relative;  
}
```

Com a adoção da variável, conseguimos deixar o nosso CSS mais dinâmico em relação à manutenção. Se um dia quisermos alterar o valor que multiplicaremos o `padding-top` e do `height`, de `16` para `10`, poderemos alterar diretamente no `variaveis.less`.