

## Faça o que eu fiz na aula

Vimos nas aulas anteriores que o Bootstrap utiliza uma unidade de medida dinâmica para, entre outras coisas, calcular espaçamentos: a medida `rem`. Mas do que se trata?

Ambas são unidades de medida que se baseiam em um tamanho de fonte-base. Vejamos o código abaixo:

```
<body>
  <div>texto</div>
</body>
```

```
body {
  font-size: 14px;
}

div {
  font-size: 2em;
}
```

Segundo este exemplo, o texto dentro de `div` terá tamanho de `2em`. O que define qual o valor de 2, nesse caso, é o valor inicial de `font-size` declarado em `body`, ou seja, `14px`. Então no exemplo acima `div` terá um valor de `font-size` de `28px` (`14px x 2em`). Basicamente, o que estamos dizendo para o CSS é que cada `em` tem um valor inicial de `14px`.

Porém o que acontece se fizermos isso?

```
<body>
<div> //font-size: 2em x 14px = 28px
  <div> //font-size: 2em x 28px = 56px
    <div> //font-size: 2em x 56px = 112px
      texto
    </div>
  </div>
</div>
</body>
```

Quando declaramos a unidade de medida como `em`, cada elemento herda o tamanho de fonte de seu elemento-pai. Isso permite que todos os elementos em que utilizamos essa medida (pode ser fonte, altura, largura, qualquer caso em que se pode aplicar unidades de medida no CSS) aumentem ou diminuam de tamanho de forma proporcional. Legal, né? Porém às vezes não queremos isso!

Nessa hora podemos usar o `rem` (*root em*, ou em raiz). Nesse caso, podemos especificar um `font-size` no elemento mais externo do nosso HTML (normalmente na tag `<html>`) e todos os elementos seguirão somente esse valor. Exemplo:

```
html {
  font-size: 14px;
}
```

```
div {  
  font-size: 2rem;  
}
```

```
<body>  
<div> //font-size: 2em x 14px = 28px  
  <div> //font-size: 2em x 14px = 28px  
    <div> //font-size: 2em x 14px = 28px  
      texto  
    </div>  
  </div>  
</div>  
</body>
```

Quando utilizamos `em` ou `rem` sem declarar um tamanho de fonte, o CSS utiliza por padrão a medida de `16px = 1em` e `16px = 1rem`. Como você já deve ter imaginado, podemos personalizar essa medida inicial de acordo com nosso projeto:

```
html {  
  font-size: 25px; //cada em ou rem equivale a 25px  
}
```

Ou:

```
html {  
  font-size: 25vw; //cada em ou rem equivale a 25% da largura do viewport  
}
```

E por aí vai. Podemos usar todas as outras unidades de medida aceitas pelo CSS (`px`, `vw`, `vh`, `%`, etc).

Voltando ao Bootstrap, as classes de `margin` e `padding` utilizam valores de 0 a 5 baseados em `rem`, de `0.25rem` até `3rem`. ou seja, como não estipulamos nada em nosso projeto, a medida-base de cada `rem` é a padrão de `16px`.