

Medidas relativas

Transcrição

Seja bem-vindo à segunda parte do curso de HTML e CSS da Alura! Nele exploraremos conceitos mais avançados desse assunto, como boas práticas e novas tecnologias que entraram nos navegadores. Para estudá-las, continuaremos com o projeto que havíamos feito no curso anterior, o site do João da Silva, o nosso desenvolvedor *front-end* fictício. Nos utilizaremos desse site para estudarmos essas novas tecnologias e recursos e boas práticas de programação.

Se você já fez o primeiro curso, já possui esse projeto. Se não, fique tranquilo, nós colocamos à sua disposição um arquivo com ele.

Píxeis e Porcentagens

A primeira página que desenvolvemos foi a da biografia do João da Silva.



Podemos notar, ao olharmos para o código CSS, que utilizamos com frequência a medida "px", ou seja, a medida *pixel*. Por exemplo:

```
.titulo-principal {
  font-size: 68px;
  text-align: center;
  background-color: #851944;
  color: #FFF;
  border-bottom: 10px solid #000;
}
```

```
.subtitulo-texto {  
  font-size: 38px;  
  clear: both;  
}
```

E assim por diante.

O que representa *1* (um) *pixel*? Se olharmos bem de perto um monitor, veremos que sua tela possui diversos micro pontos, estes são os *píxeis*.

No exemplo acima, passamos que o título principal da página deve ter uma fonte de *68 píxeis* de tamanho. Tal medida é muito boa, porque nos permite um controle dos tamanhos dos elementos. Porém não é apenas essa que estamos usando. Também temos a porcentagem, como em:

```
body {  
  font-family: "Crimson Text", "Times New Roman", serif;  
  background-color: #F2FFFC;  
  font-size: 120%;  
  line-height: 1.5;  
}
```

Utilizamos a porcentagem para termos um controle do tamanho da fonte da página com base no tamanho padrão da fonte do navegador.

Outro caso do uso de porcentagem é para definir larguras. Por exemplo:

```
main {  
  width: 86%;  
}
```

Se você baixar o arquivo do projeto, pode ser que essas porcentagens estejam diferentes. Isso acontece porque elas devem variar de monitor para monitor. Então dê uma modificada nelas para adaptá-las ao seu. Mais para frente veremos que existe um meio de não precisar fazer isso.

O tamanho das fontes ou das larguras dos objetos irá variar, ao utilizarmos a porcentagem, de acordo com o navegador e o monitor que estamos usando, ou seja, o contexto em que a fonte se insere.

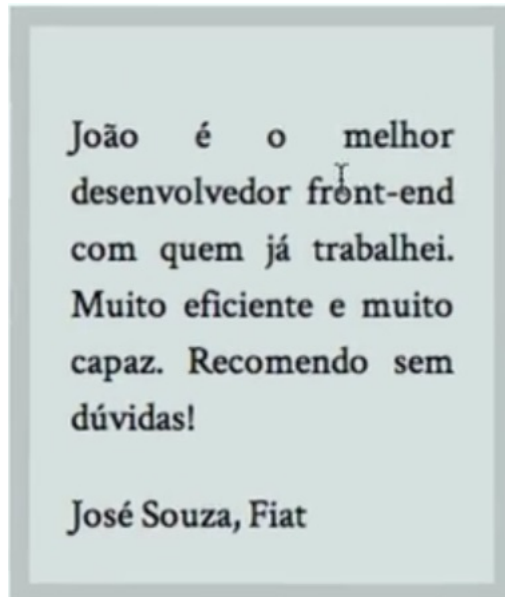
Se inspecionarmos o elemento *aside*:



Observamos:

```
.navegacao-site {  
  width: 14%;  
}
```

Ele possui uma largura de 14% da tela. Mas será que é sempre assim? Vamos pegar outro caso, o do *blockquote*:



Observamos:

```
.citacao-bio {  
  ...  
  padding: 20px;  
}
```

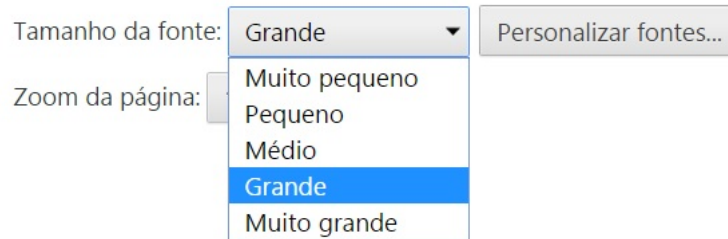
Se mudarmos de 20 px para 10%, será que será 10% da tela? Se inspecionarmos os tamanhos dos elementos:



Perceba que o *padding* está com 72 pixels. Como nossa tela não tem 720 pixels, é muito maior, esses 10% não estão relacionados a ela, mas sim ao *container*, ao pai do elemento que se quer modificar o tamanho. E lembre-se que a relação é sempre entre as larguras.

Voltando aos pixels, nós temos um pequeno problema ao trabalhar com essa medida. O próprio usuário pode mudar o tamanho da fonte utilizando as configurações do navegador:

Conteúdo da web



Se voltarmos à nossa página, verificamos que o texto principal ficou maior, porém título e subtítulos mantiveram seu tamanho original. Isso acontece porque o texto principal está definido por uma porcentagem, enquanto que o título e subtítulos estão estritamente definidos pelo número de píxeis. Não importa o quanto você modifique o tamanho da fonte nas configurações. Se o tamanho do elemento for definido por píxeis, ele manterá esse tamanho.

O que podemos fazer é nos utilizar do mesmo recurso do *body*:

```
.titulo-principal {  
    font-size: 300%;  
}
```

A medida *rem*

Podemos aplicar as porcentagens em todos os elementos, mas sempre lembrando que para alguns ela está relacionada à largura do elemento pai. Isto pode causar alguma confusão, uma vez que teríamos que aplicar uma porcentagem em cima de outra, no caso do *padding*.

Utilizaremos, então, uma nova medida que usa o tamanho da fonte como base, o ***rem***:

$1,25\text{ rem} = 1,25 \times \text{tamanho da fonte}$

Por exemplo, se a fonte do navegador for de 20px, temos:

$20\text{px} \times 1,25 = 25\text{px}$

O *rem* é uma medida muito boa para quando quisermos que os elementos sigam o padrão do texto principal.

A medida *ch*

Voltando ao *blockquote*, vemos que seus elementos filhos são baseados na sua largura. Então não faz sentido utilizar o *rem* aqui, uma vez que este é baseado na altura da fonte.

Então vamos utilizar uma medida que facilita principalmente na edição de textos dentro de caixas, a medida ***ch***:

$1\text{ ch} = \text{largura do caractere "zero" da fonte utilizada}$

Modificando o tamanho da fonte do navegador, esta medida também modifica-se. Essa medida é boa porque não importa muito a fonte que você está utilizando, ela se adequará muito bem.

A medida *em*

Agora vamos tentar aumentar o tamanho da fonte do *blockquote*:

```
.citacao-bio {  
  ...  
  font-size: 200%;  
}
```

Ao fazermos isso, nada acompanha proporcionalmente esse aumento. Seria interessante que todos os elementos modificassem de tamanho com base na fonte do elemento pai.

Para isso, utilizamos a medida **em** que indica o tamanho da fonte do elemento, não do navegador como é o caso do *rem*.

Claro que se aumentarmos a fonte do navegador, como todos os elementos estão ligados a ela, o *blockquote* e seus elementos irão ficar ainda maiores.

Font-size no HTML

No *body* sua *font size* era de 120%. Se aumentarmos para 220%, verifique que o tamanho da borda não aumentou. Isso acontece porque a fonte que padroniza tudo não é a do *body*, mas sim a do navegador, aquela que modificamos nas configurações.

Porém conseguimos modificar com base no elemento HTML, declarando:

```
html {  
  font-size: 200%;  
}
```

Se agora voltarmos ao título principal e declararmos:

```
.titulo-principal {  
  ...  
  border-bottom: 0.5rem  
}
```

A borda fica mais larga.

Então conseguimos controlar o *rem* tanto pela configuração do navegador quanto pelo *font-size* do HTML (não do *body*). Tanto que é melhor excluirmos o *font-size* do *body* para que toda declaração de *rem** faça sentido.

Resumindo

Vimos, então, três novas medidas:

- *rem*: tem como padrão a fonte do navegador;
- *em*: tem como padrão a fonte do elemento pai;
- *ch*: tem como base a largura do caractere zero da fonte usada;

Vimos que a porcentagem tem como base ou a largura do elemento pai ou o *font-size* do elemento anterior.

Utilizando essas medidas conseguimos um CSS mais flexível, principalmente se quisermos manter um padrão de tamanho de fonte. Utilizando o *font-size* do HTML, é apenas ali que será necessário fazer alterações.