0.1 Estilizando Com CSS (cont.)

Nesta seção continuaremos com algumas propriedades básicas que se pode utilizar com o CSS.

0.1.1 Imagem de Fundo

A propriedade *background-image* permite indicar um arquivo de imagem para ser exibido ao fundo do elemento. Por exemplo:

```
1
2 h1 {
3   background-image: url(yellow.png);
4 }
```

Através da declaração "url"o navegador consegue requisitar o arquivo yellow.png. É importante destacar que url aceita quaqluer caminho conhecido da imagem.

0.1.2 Mais sobre Bordas

Quando se define o estilo de uma board utilizando border-style, é possível definir outros estilos além do "solid"mostrado anteriormente na aula 2. Alguns estilos possíveis são: *dashed*, *dotted*, *double* e *groove*. Existem outros estilos, cabendo a vocês olharem a documentação e testarem como exercício todos as possibilidades de estilos.

0.1.3 Cores

Propriedades como background-color, color, border-color, entre outras aceitam uma cor como valor. Existem várias maneiras de definir cores quando utilizamos o CSS. A primeira e mais simples é usando o nome da cor:

```
1
2 h1 {
3   color: red;
4 }
5
6 h2 {
7   background-color: yellow;
8 }
```

O difícil é acertar a exata variação de cor que queremos no design. Por isso, é bem incomum usarmos cores com seus nomes. O mais comum é

definir a cor com base em sua composição **RGB**. RGB é um sistema de cor bastante comum aos designers. Ele permite especificar até 16 milhões de cores com uma combinação de três cores base: *Vermelho (Red)*, *Verde (Green)*, *Azul (Blue)*. Podemos escolher a intensidade de cada um desses três canais básicos, numa escala de 0 a 255. Um amarelo forte, por exemplo, tem 255 de Red, 255 de Green e 0 de Blue (255, 255, 0). Se quiser um laranja, basta diminuir um pouco o verde (255, 200, 0). E assim por diante. No CSS, podemos escrever as cores tendo como base sua composição RGB. No CSS3 - que veremos melhor depois (se der tempo) - há até uma sintaxe bem simples pra isso:

```
1
2 h1 {
3   color: rgb(255, 200, 0);
4 }
```

Essa sintaxe funciona nos browsers mais modernos mas não é a mais comum na prática, por questões de compatibilidade. O mais comum é a notação hexadecimal, ao escrevermos, por exemplo: #F2EDED. Essa sintaxe tem suporte universal nos navegadores e é mais curta de escrever. Na notação hexadecimal (que começa com #), temos 6 caracteres. Os primeiros 2 indicam o canal Red, os dois seguintes, o Green, e os dois últimos, Blue. Ou seja, RGB. Na base hexadecimal, os algarismos vão de zero a quinze (ao invés do zero a nove da base decimal comum). Para representar os algarismos de dez a quinze, usamos letras de A a F. Nessa sintaxe, portanto, podemos utilizar números de 0-9 e letras de A-F.

0.1.4 Espaçamento, Margem e Dimensões

Este é, talvez, um dos tópicos mais importantes sobre CSS em relação ao poder da estilização, e o que mais atrapalha, deixando os designers aflitos. Trabalhar com essas propriedades requer prática, pois se aplicadas erradas podem levar há uma má formatação da página.

Utiliza-se a propriedade *padding* para **espaçamento**, *margem* para **margem**, *height* e *width* para alterar dimensões dos elementos.

Padding

A propriedade padding é utilizada para definir um espaçamento interno em alguns elementos (por espaçamento interno queremos dizer a distância entre o limite do elemento, sua borda, e seu respectivo conteúdo) e tem as subpropriedades listadas a seguir:

- padding-top
- padding-right
- padding-bottom
- padding-left

Essas propriedades aplicam uma distância entre o limite do elemento e seu conteúdo acima, à direita, abaixo e à esquerda respectivamente. Essa ordem é importante para entendermos como funciona a *shorthand property* do padding. Se passado somente um valor para a propriedade padding, esse mesmo valor é aplicado em todas as direções.

```
1
2 p {
3   padding: 10px;
4 }
```

Se passados dois valores, o primeiro será aplicado acima e abaixo (equivalente a passar o mesmo valor para padding-top e padding-bottom) e o segundo será aplicado à direita e à esquerda (equivalente ao mesmo valor para padding-right e padding-left):

```
1
2 p {
3   padding: 10px 15px;
4 }
```

Se passados três valores, o primeiro será aplicado acima (equivalente a padding-top), o segundo será aplicado à direita e à esquerda (equivalente a passar o mesmo valor para padding-right e padding-left) e o terceiro valor será aplicado abaixo do elemento (equivalente a padding-bottom):

```
1
2 p {
3   padding: 10px 20px 15px;
4 }
```

Se passados quatro valores, serão aplicados respectivamente a padding-top, padding-right, padding-bottom e padding-left. Para facilitar a memorização dessa ordem, basta lembrar que os valores são aplicados em *sentido horário*.

```
1
2 p {
3   padding: 10px 20px 15px 5px;
4 }
```

Margin

A propriedade margin é bem parecida com a propriedade padding, exceto que ela adiciona espaço após o limite do elemento, ou seja, é um espaçamento além do elemento em si. Além das subpropriedades listadas a seguir, há a *shorthand property* margin que se comporta da mesma maneira que a *shorthand property* do padding vista na seção anterior.

- margin-top
- margin-right
- margin-bottom
- margin-left

Há ainda uma maneira de permitir que o navegador defina qual será a dimensão da propriedade padding ou margin conforme o espaço disponível na tela: definimos o valor auto para os espaçamentos que quisermos. No exemplo a seguir, definimos que um elemento não tem nenhuma margem acima ou abaixo de seu coonteúdo e que o navegador define uma margem igual para ambos os lados de acordo com o espaço disponível:

```
1
2 p {
3   margin: 0 auto;
4 }
```

Dimensões

É possível determinar as dimensões de um elemento através do tamanho (height) e do comprimento (width):

```
1
2 p {
3   background-color: red;
4   height: 400px;
5   width: 400px;
6 }
```

0.1.5 Unidades de Medida

Acredito que vocês perceberam que só tenho utilizado pixels (px) como unidade de medida nos exemplos. Porém, há inúmeras unidades, e acreditem, elas podem ser aplicadas aos mais diversos tipos de problemas. Assim, podemos classificar as medidas CSS em dois grandes grupos conforme mostrados a seguir:

Unidades de medidas de distância

As unidades de medida de distância definem medidas na horizontal, na vertical ou em qualquer direção.

A sintaxe para se declarar o valor de uma unidade de medida de distância é composta por um número inteiro ou decimal imediatamente precedido do sinal '+' (mais) ou do sinal '-' (menos) — o sinal '+' (mais) é o padrão — imediatamente seguido por uma unidade de medida CSS válida (por exemplo: px, em, deg, s, dpi, vw, etc).

A unidade de medida é opcional quando se declara o valor '0' (zero).

Algumas das propriedades CSS permitem que sejam declarados valores negativos para unidades de medida.

Medidas Relativas São medidas calculadas em relação a uma outra unidade de medida. Usar unidades de medidas relativas é mais apropriado para se obter ajustes em diferentes tipos de mídia. (por exemplo: ajustar de uma tela de monitor para uma impressora laser).

O valor é tomado em relação:

- **em**: ...ao tamanho da fonte ('font-size') do elemento no qual a unidade é declarada;
- **ex**: ...a altura da letra x (xis) da fonte herdada;
- **ch**: ...a largura acrescida do espaçamanto em volta do número 0 (zero) da fonte do elemento no qual a unidade é declarada;
- **rem**:...ao tamanho da fonte ('font-size') do elemento raiz do documento.

O valor é igual a:

- **vw**: 1% da largura da viewport;
- **vh**: 1% da altura da viewport;

- **vmin**: a menor medida entre vw e vh;
- **vmax**: a maior medida entre vw e vh.

A especificação do W3C classifica tecnicamente a porcentagem (%) como tipo de dado e não unidade de medida.

Medidas Absolutas São medidas que não estão referenciadas a qualquer outra unidade. São as unidades de medida de comprimento definidas nos sistemas de medidas pela física, tais como, centímetro, polegada, etc... e a unidade pixel para o ângulo visual. São indicadas para serem usadas quando as mídias de exibição são perfeitamente conhecidas.

- **cm - centímetro**: 96px/2.54;
- **mm - milímetro**: 1/10cm;
- **q - 1/4 do milímetro**: 1/40cm;
- **in - polegada**: 2,54cm = 96px;
- **pc - pica (é assim mesmo :X)**: 12 points ou 1/6in;
- **pt - point**: 1/72in;
- **px - pixel**: 1/96in.

Unidades de medidas para outras quantidades

Unidades para ângulos

- **deg - grau**: um círculo tem 360deg;
- **grad - grado**: um círculo tem 400grad;
- **rad - radiano**: um círculo tem 2πrad;
- **turn - volta**: um círculo tem 1turn;

Unidades para resolução

- **dpi - dots per inch**: pontos por polegada;
- **dpcm - dots per centimeter**: pontos por centímetro;
- **dppx - dots per pixel**: pontos por pixel;

Acreditem, nem sempre é recomendável utilizar apenas pixels como unidade de medida. Observem a seguir exemplos de declaração usando algumas medidas:

```
1
2  div { margin: 1.5em; }
3  h4 { margin: 2ex; }
4  p { font-size: 14px; }
5  .classe { padding: 90%; }
6  hr { width: 14pt; }
7  h1 { margin: 1pc; }
8  h2 { font-size: 4mm; }
9  p.classe { padding: 0.3cm; }
10 h5.classe { padding: 0.5in; }
11 @media ( min-resolution: 2dppx ) { ... }
12 background-image: linear-gradient( 45deg, white,
       black );
13 div { width: 50vw; }
```

Referências Bibliográficas

[1] HTML5 e CSS3: Domine a web do futuro - Lucas Mazza, editora Casa do Código.

[2] Desenvolvimento Web com HTML, CSS e JavaScript - Caelum ensino e inovação, Curso WD-43.

[3] HTML Documentação: `https://devdocs.io/html/`

[4] W3C – CSS documentação: `https://www.w3.org/Style/CSS/#specs`