CSS

Introdução

Agora que aprendemos as principais tags HTML, vamos aprender a controlar o aspecto visual das nossas páginas web! Embora seja possível "embutir" esses atributos de estilo nas tags HTML, esta não é a prática recomendada pela W3C, além de não ser uma solução prática caso seja necessário alterar algum desses atributos depois.

Vamos criar arquivos CSS separados das nossas páginas, inserindo o <link href="arquivo.css" rel="stylesheet"> dentro da tag HEAD, pois ela vai controlar a aparência dos elementos do BODY, mas o que está escrito dentro delas não será diretamente exibido na página web, e sim interpretado pelo navegador para que ele possa renderizar nossa página da forma que gostaríamos.

O HTML não deve conter tags para formatar textos, mas sim para definir o conteúdo de um documento. Por exemplo:

```
<h1>Cabeçalho da página</h1>
<p>Texto de um parágrafo. Este é um parágrafo. Quero pular uma
linha. <br> Pulei uma linha. </p>
```

O W3C não recomenda o uso da tag <font>, e mesmo que os navegadores ainda interpretem as informações nela contidas, o uso de FONT já caiu em desuso. A forma correta de fazer toda a formatação de páginas web é utilizando CSS, utilizando um arquivo CSS externo.

Uma regra em CSS consiste na combinação de um seletor com um bloco de declarações. Por exemplo: quero que o plano de fundo da minha página seja o arquivo foto.jpg. A regra em CSS será:

```
body {background-image: url("foto.jpg");}
```

Neste exemplo, body é o nome do seletor, e o que está entre chaves é uma declaração. Caso eu também queira que o arquivo foto.jpg cubra toda a página, devo reescrever a regra da seguinte forma:

```
body {background-image: url("foto.jpg"); background-size:
cover;}
```

Observe que agora o bloco entre chaves possui duas declarações. Não se esqueça de colocar o ponto-vírgula (;) ao final de cada uma das declarações. O CSS também não faz diferenciação de quebra de linha, então é possível pular linhas para deixar o código mais organizado:

```
body {
background-image: url("foto.jpg");
background-size: cover;
}
```

É possível incluir informações de estilo em uma página utilizando um arquivo .css externo. Para isso, é necessário inserir o seguinte código no arquivo HTML (também entre as tags <head> e </head> :

```
<link href="style.css" rel="stylesheet"/>
```

Seletores

Seletores CSS são utilizados para referenciar os elementos HTML que serão modificados, com base no id, classe, atributo, nome do elemento, ou mais.

Seletores de elemento são baseados nos nomes dos elementos, isto é, nas tags de HTML. Portanto, todas as tags HTML que aprendemos até agora podem ser utilizadas como seletores: BODY, P, DIV, H1, TABLE, TD, TH, TR, IMG, A, FORM, INPUT, SELECT, TEXTAREA etc. Chamadas CSS nos seletores de elemento afetam todos os elementos marcados com a mesma tag.

Atenção:As tags HTML, HEAD e TITLE não devem ser utilizadas como seletores em CSS.

O exemplo abaixo mostra como alterar o estilo de todos os parágrafos de uma página, centralizando o texto e colocando-o na cor vermelha.

```
p {
text-align: center;
color: red;
}
```

Seletores de ID utilizam o atributo ID de uma tag para referenciar elementos específicos. O ID de um elemento deve ser único dentro da página! Portanto, esse seletor é utilizado para apenas um elemento. Por exemplo, para alterar a aparência de apenas um parágrafo, é necessário dar um ID a ele e referenciar esse ID antecedido pelo símbolo # no arquivo .css.

O exemplo abaixo altera o estilo do parágrafo com ID "p1".

```
#p1 {
text-align: justify;
color: blue;
}
```

Seletores de classe utilizam o atributo de classe para referenciar um conjunto de elementos específicos. A diferença entre classe e ID: enquanto o ID é único, a classe afeta um grupo de elementos. Portanto, o atributo classe (class) pode se repetir na mesma página. Para alterar a aparência de um conjunto de parágrafos, é necessário atribuir uma classe (class) a cada um deles e referenciar a class antecedido pelo símbolo . no arquivo .css.

O exemplo abaixo altera o estilo de todos os elementos com o atributo CLASS igual a "info".

```
.info {
text-align: center;
color: red;
}
```

Também é possível especificar quais elementos serão afetados por uma declaração de classe. Se eu quiser que apenas as DIV com a classe "info" sejam afetadas, a declaração deve ficar assim:

```
div.info {
text-align: center;
color: black;
}
```

Se você possuir mais de um elemento com a mesma definição de estilo, é possível agrupar as declarações. Por exemplo, se todos os cabeçalhos H1, H2 e H3 possuem letra vermelha, a declaração CSS pode ser feita assim:

```
h1, h2, h3 {
color: red;
}
```

Comentários

Comentários em CSS são feitos entre /* e */

```
h1, h2, h3 {
color: red; /*cor do h1, h2 e h3 é vermelha*/
}
```

Cores

Cores podem ser especificadas de algumas maneiras diferentes:

- Utilizando nomes pré-definidos (red, orange, pink, green, tomato etc). Há 140 nomes pré-definidos .
- Valores em hexadecimal: sequência de 6 dígitos de 0 a F, sempre antecedidos por #. Exemplo: #000000 corresponde ao preto e #FF0000, ao vermelho.
- HSL (hue, saturation, lightness): combinação de matiz, saturação e luminosidade. Por exemplo, hsl(0,100%,50%) corresponde ao vermelho.
    - Matiz/Hue varia entre 0 e 360 (0 = vermelho; 120 = verde; 240 = azul)
    - Saturação é uma porcentagem, que varia de 0% (tom de cinza) a 100% (cor)
    - Luminosidade também é uma porcentagem, com 0% sendo preto e 100%, branco.
- HSLA: combinação de matiz, saturação, luminosidade e alfa (transparência). O alfa varia de 0 (totalmente transparente) a 1 (totalmente opaco). Números decimais correspondem a cores translúcidas. Por exemplo, hsl(0,100%,50%,0.5) corresponde ao vermelho 50% opaco.
- RGB: combinação de vermelho, verde e azul. As cores variam de 0 a 255. Por exemplo, vermelho é rgb(255,0,0)
- RGBA: combinação de vermelho, verde, azul e alfa (transparência). As cores variam de 0 a 255, o alfa varia de 0 (totalmente transparente) a 1 (totalmente opaco). Números decimais correspondem a cores translúcidas. Vermelho com 50% de opacidade corresponde a rgba(255,0,0,0.5)

É possível modificar a cor de fundo de qualquer elemento com a declaração background-color

```
h1 {
background-color: #FF3399;
}
```

É possível modificar a cor do texto de qualquer elemento com a declaração color

```
h2 {
color: Tomato;
}
```

É possível modificar a aparência da borda de qualquer elemento com a declaração border, desde que na seguinte ordem: espessura (em pixels), estilo (solid - normal, dashed - tracejada e dotted - pontilhada) e cor.

```
p{
border: 2px solid #FF3399;
}
```

Planos de fundo

Planos de fundo podem ser cores, como vimos acima, mas também podem conter imagens. Para que isso ocorra, é necessário utilizar a declaração background-image.

```
body{
background-image: url("imgFundo.jpg");
}
```

As imagens de fundo, por padrão, se repetem tanto horizontalmente quanto verticalmente. Isso pode ser controlado pela declaração background-repeat.

```
body{
background-image: url("imgFundo.jpg");
background-repeat: repeat-x;
}
```

O exemplo acima irá repetir a imagem de fundo apenas horizontalmente. Para repetir apenas verticalmente, utiliza-se repeat-y. Para não repetir, utilizamos no-repeat.

Também é possível demarcar a posição que a imagem de fundo ficará na página, tanto horizontalmente quanto verticalmente, com background-position.

```
body{
background-image: url("imgFundo.jpg");
background-repeat: repeat-x;
background-position: center top;
}
```

No exemplo acima, a imagem estará centralizada horizontalmente, mas no topo da página. Horizontalmente, ela poderá se posicionar à direita (right), esquerda (left) ou centro (center). Verticalmente, ela poderá se posicionar acima (top), abaixo (bottom) ou no centro (center).

Para que a imagem de fundo não role conforme a rolagem da página, utilizamos a declaração background-attachment com o valor fixed.

```
body{
background-image: url("imgFundo.jpg");
background-repeat: repeat-x;
background-position: center top;
background-attachment: fixed;
}
```

Margens

Margens são espaços criados ao redor dos elementos, isto é, fora das bordas definidas. Margens podem ser definidas com valores absolutos (em pixels, cm, pt ou outras unidades), valores percentuais, ou até automaticamente (auto), isto é, calculadas pelo navegador.

É possível atribuir valores negativos às margens.

```
div{
margin-top:15px;
margin-bottom: 20px;
margin-right: 30px;
margin-left: 30px;
}
```

Padding

Padding (enchimento) gera espaço ao redor dos conteúdos de um elemento, isto é, dentro das bordas definidas. Esses valores também podem ser especificados em qualquer unidade. Ao contrário das margens, o padding não aceita valores negativos.

```
p{
padding-top:10px;
padding-bottom: 15px;
padding-right: 100px;
padding-left: 100px;
}
```

Padding, margin e border podem ser também comandos *shorthand*, isto é, comandos de escopo reduzido que definem todos os padding, margin e border de uma só vez, sempre em sentido horário, iniciando do topo.
O exemplo acima pode ser reescrito da seguinte forma:

```
p{
padding:10px 100px 15px 100px;
}
```

Tamanho

O tamanho dos elementos é controlado pela altura (height) e largura (width). Esses valores também podem ser especificados em qualquer unidade.

```
img.foto{
height: 100px;
width: 100px;
}
```

Além disso, existem propriedades como max-height e max-width, que estipulam altura e largura máximas para um elemento. Isso é muito importante para criar páginas que se ajustam a telas de tamanhos diferentes. Da mesma forma, existem min-height e min-width, que estipulam altura e largura mínimas.

Formatação de texto

Além da cor, é possível formatar textos de outras formas. Por exemplo, modificando o alinhamento do texto com text-align ou modificando a decoração do texto com text-decoration. Normalmente, o text-decoration é utilizado para retirar o sublinhado de links em alguns sites.

```
p{
text-align: left;
text-decoration: none;
}
```

O texto pode ser alinhado à esquerda (left), direita (right), ao centro (center) ou justificado (justify).

Para modificar a fonte de um texto, utiliza-se o font-family. A propriedade font-style pode deixar o texto normal ou itálico (italic). Para controlar o tamanho da fonte, utiliza-se font-size. A propriedade font-weight controla o peso da fonte (bold = negrito).

```
p{
font-family: "Times New Roman", serif;
font-style: italic;
font-size: 15px;
font-weight: bold;
}
```

Links

É possível alterar o CSS de links com todas as propriedades já mencionadas anteriormente: cor, fundo, fonte etc.

Além disso, é possível controlar a aparência dos links de acordo com os estados deles. São 4 estados possíveis, que devem aparecer nessa exata ordem:

- a:link - link normal que ainda não foi visitado pelo usuário
- a:visited - link que foi visitado pelo usuário
- a:hover - quando o mouse passa por cima do link
- a:active - link no momento em que é clicado

```
a:link{
font-weight: bold;
}

a:visited{
font-weight: bold;
color: #FF3399;
}

a:hover{
font-weight: bold;
text-decoration: overline;
}

a:active{
color: tomato;
```

```
background-color: white;
}
```

Posicionar elementos

O atributo float é utilizado para posicionar elementos na página. Ele pode assumir três valores: left (esquerda), right (direita) ou none (sem efeito). É possível utilizá-lo para definir a posição de DIVs na página, assim como para outros propósitos.

Para centralizar elementos, podemos utilizar margin: auto (só tem efeito visualemte se o elemento possuir uma declaração width com valor inferior a 100%).

Box Model

Todos os elementos HTML podem ser considerados como retângulos (ou caixas). O termo "Box Model" é utilizado quando falamos de projetar layouts e se refere ao espaço que um elemento verdadeiramente ocupa.

Por exemplo, criar um div com 300px de largura significa que todo o conteúdo dentro daquela div terá 300px de largura. Ao acrescentar propriedades como Padding (enchimento), Border (borda) e Margin (margem) nessa mesma div, suas dimensões se modificarão. Padding e margens sempre são transparentes.

No exemplo abaixo, o parágrafo possui borda de 1px e 0 de margem. O elemento abaixo dele é uma div com 20px de margem (observe que ela "começa" um pouco depois do parágrafo), 10px de borda rosa e 20px de enchimento/padding. A largura da div é de 300px, mas na realidade ela ocupa 460px. Pode ser feito o mesmo cálculo para a altura, que não foi especificada na div, mas independente disso, terá mais 160px do que o espaço do conteúdo.

Margem + borda + padding (esquerda)	Elemento	Margem + borda + padding (direita)	Total
20 + 10 + 50 = 80px	300px	20 + 10 + 50 = 80px	80 + 300 + 80 = 460px

Box Model

Todos os elementos HTML podem ser considerados como retângulos (ou caixas). O termo "Box Model" é utilizado quando falamos de projetar layouts e se refere ao espaço que um elemento verdadeiramente ocupa.

Este texto é o conteúdo da caixa. Acrescentei padding de 50px, margem de 20px e borda de 10px. Agora veremos frases em latim. Lorem ipsum dolor sit amet, ut eam tempor feugiat molestiae. Esse accommodare at eum, sanctus constituto quo id. Ex has brute everti, ne aperiam fuisset disputando quo. Ei idque consequat has, his falli iuvaret principes eu, perpetua consectetuer has te. Quo recusabo scripserit et. Option signiferumque vis eu.