0.1 Estilizando com CSS

Quando escrevemos o HTML, marcamos o conteúdo da página com tags que melhor representam o significado daquele conteúdo. Aí quando abrimos a página no navegador é possível perceber que ele mostra as informações com estilos diferentes. Um h1, por exemplo, fica em negrito numa fonte maior. Parágrafos de texto são espaçados entre si, e assim por diante. Isso quer dizer que o navegador tem um estilo padrão para as tags que usamos. Antigamente, isso era feito no próprio HTML. Se quisesse um título em vermelho, era só fazer:

```
1
2 <h1><font color="yellow">Yellow Ledbetter</font></h1>
```

Além da tag <font>, várias outras tags de estilo existiam. Contudo, tags HTML para estilo são má prática hoje em dia e jamais devem ser usadas. Em seu lugar, surgiu o CSS, que é uma outra linguagem, separada do HTML, com objetivo único de cuidar da estilização da página. A vantagem é que o CSS é bem mais robusto que o HTML para estilização, como veremos.

Entretanto, escrever formatação visual misturado com conteúdo de texto no HTML se mostrou algo bem impratiçavel. O CSS resolve isso separando as coisas; regras de estilo não aparecem mais no HTML, apenas no CSS.

0.2 Sintaxe do CSS

A sintaxe do CSS tem estrutura simples: é uma declaração de um elemento que recebe um conjnto de propriedades e valores separados por um sinal de dois pontos ":", e cada propriedade é separada por um sinal de ponto e vírgula ";"da seguinte maneira:

```
1
2 [elemento] {
3   propriedade: valor;
4 }
```

Além do elemente, podemos incluir outros meios possíveis de referência para renderização. Contudo, iremos ver no decorrer das aulas.

0.3 Métodos de Inclusão do CSS

Existem três formas de se trabalhar com CSS juntamente com o HTML. Irei explicar os três métodos e qual destes é recomendado como boa prática.

0.3.1 Método Inline

O método inline tem como característica a definição de propriedades do CSS dentro da propriedade style que todo elemente HTML possui. Por exemplo:

```
1
2 <h1 style="color: yellow;">Yellow Ledbetter</h1>
```

O texto acima estamos modificando a cor do título usando a propriedade color do CSS. Entretanto, foi dito anteriormente que uma das grandes vantagens do CSS era manter as regras de estilo fora do HTML. Logo, esta abordagem não é aconselhável e deve ser evitada ao máximo.

0.3.2 Incorporado

A outra maneira de se utilizar o CSS é declarando suas propriedades dentro de uma tag <style>. Como estamos declarando as propriedades visuais de um elemento em outro lugar do nosso documento, precisamos indicar de alguma maneira a qual elemento nos referimos. Fazemos isso utilizando um **seletor CSS**. É basicamente uma forma de buscar certos elementos dentro da página que receberão as regras visuais que queremos. No exemplo a seguir, usaremos o seletor que pega todas as tags h1 e altera sua cor:

```
1
2  <!DOCTYPE html>
3  <html>
4    <head>
5      <meta charset="utf-8">
6      <title>Pearl Jam</title>
7      <style>
8        h1 {
9          color: yellow;
10       }
11     </style>
12   </head>
13   <body>
14     <h1>Yellow Ledbetter</h1>
15   </body>
16 </html>
```

A tag <style> deve estar sempre dentro da declaração do <head> do documento. Apesar deste método já separar o CSS das tags HTML, ainda há uma forte dependência e uma certa mistura de padrões. Afinal, queremos que nosso documento HTML apenas se preocupe com suas marcações.

0.3.3 Arquivo Externo

A terceira maneira de declararmos os estilos do nosso documento é com um arquivo externo, geralmente com a extensão ".css". Para que seja possível declarar nosso CSS em um arquivo à parte, precisamos indicar em nosso documento HTML uma ligação entre ele e a folha de estilo (arquivo com a extensão ".css"). Além da melhor organização do projeto, a folha de estilo externa traz ainda as vantagens de manter nosso HTML mais limpo e do reaproveitamento de uma mesma folha de estilos para diversos documentos. A indicação de uso de uma folha de estilos externa deve ser feita dentro da tag <head> do documento HTML:

```
1
2  <!DOCTYPE html>
3  <html>
4    <head>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href="estilos.css">
7    </head>
8    <body>
9      <h1>Yellow Ledbetter</h1>
10   </body>
11 </html>
```

E dentro do arquivo estilos.css colocamos apenas o conteúdo do CSS:

```
1
2  h1 {
3    color: yellow;
4  }
```

lembrando que a tag <link> deve sempre referenciar o exato local onde está o arquivo .css.

0.4 Estrutura dos Arquivos de Um Projeto

Como todo tipo de projeto de software, existem algumas recomendações quanto à organização dos arquivos de um site. Não há nenhum rigor técnico quanto a essa organização e, na maioria das vezes, você vai adaptar as recomendações da maneira que for melhor para o seu projeto. Como um site é um conjunto de páginas Web sobre um assunto ou qualquer outra coisa, é comum todos os arquivos de um site estarem dentro de uma só pasta e, assim como um livro, é recomendado que exista uma "capa", uma página inicial que possa indicar para o visitante quais são as outras páginas que

fazem parte desse projeto e como ele pode acessá-las, como se fosse o índice do site. Assim, o projeto que vocês iniciaram no exercício deverá obedecer esta regra daqui por diante. Organização contará como critério de avaliação. A organização que vou pedir a vocês é a seguinte:

```
1
2  dir: <nome_do_projeto>
3    subdir: <css>
4    subdir: <img> (para imagens>
5    subdir: <js>
6    subdir: <views> (para outras paginas que nao o
       indice)
7    index.html (pagina principal do site - nome a sua
       escolha)
```

0.5 Propriedades Tipográficas e Fontes

Da mesma maneira que alteramos cores, podemos alterar o texto. Podemos definir fontes com o usoda propriedade font-family. A propriedade font-family pode receber seu valor com ou sem aspas. No primeiro caso, passaremos o nome do arquivo de fonte a ser utilizado, no último, passaremos a família da fonte. Por padrão, os navegadores mais conhecidos exibem texto em um tipo que conhecemos como "serif". As fontes mais conhecidas (e comumente utilizadas como padrão) são "Times"e "Times New Roman", dependendo do sistema operacional. Elas são chamadas de fontes serifadas pelos pequenos ornamentos em suas termiinações. Podemos alterar a família de fontes que queremos utilizar em nosso documento para a família "sans- serif"(sem serifas), que contém, por exemplo, as fontes "Arial"e "Helvetica". Podemos também declarar que queremos utilizar uma família de fontes "monospace"como, por exemplo, a fonte "Courier".

```
1
2  h1 {
3    font-family: serif;
4  }
5
6  h2 {
7    font-family: sans-serif;
8  }
9
10 p {
11   font-family: monospace;
```

```
12 }
```

Existe uma outra forma de declarar um conjunto de fontes. Por exemplo, podemos aplicar ao elemento body as fontes que desejamos, e então este será aplicado a todo o documento:

```
1
2 body {
3   font-family: "Arial", "Helvetica", sans-serif;
4 }
```

Nesse caso, o navegador verificará se a fonte "Arial"está disponível e a utilizirá para renderizar os textos de todos os elementos do nosso documento que, por cascata, herdarão essa propriedade do elemento body.Caso a fonte "Arial"não esteja disponível, o navegador verificará a disponibilidade da próxima fonte declarada, no nosso exemplo a "Helvetica". Caso o navegador não encontre também essa fonte, ele solicita qualquer fonte que pertença à família "sans-serif", declarada logo a seguir, e a utiliza para exibir o texto, não importa qual seja ela.

0.6 Alinhamento e Decoração do Texto

Uma das propriedades mais simples, porém muito utilizada, é a que diz respeito ao alinhamento de texto: a propriedade text-align.

```
1
2 p{
3   text-align: right;
4 }
```

O exemplo anterior determina que todos os parágrafos da nossa página tenham o texto alinhado para a direita. Também é possível determinar que um elemento tenha seu conteúdo alinhado ao centro ao definirmos o valor *center* para a propriedade text-align, ou então definir que o texto deve ocupar toda a largura do elemento aumentando o espaçamento entre as palavras com o valor *justify*. O padrão é que o texto seja alinhado à esquerda, com o valor *left*, porém é importante lembrar que essa propriedade propaga-se em cascata.

É possível configurar também uma série de espaçamentos de texto com o CSS:

```
1
2 p {
3
4   line-height: 3px; /* tamanho da altura de cada
       linha */
```

```
5   letter-spacing: 3px; /* tamanho do espaco entre
        cada letra */
6   word-spacing: 5px; /* tamanho do espaco entre cada
        palavra */
7   text-indent: 30px; /* tamanho da margem da primeira
        linha do texto */
8 }
```

0.7 Bordas

As propriedades do CSS para definirmos as bordas de um elemento nos apresentam algumas opções. Podemos, para cada borda de um elemento, determinar sua cor, seu estilo de exibição e sua largura. Por exemplo:

```
1
2 body {
3   border-color: red;
4   border-style: solid;
5   border-width: 1px;
6 }
```

A propriedade border tem uma forma resumida para escrever os mesmos estilos que adicionamos acima, mas de uma maneira mais simples:

```
1
2 body {
3   border: 1px solid red;
4 }
```

Para que o efeito da cor da borda tenha efeito, é necessário definir qualquer valor em *border-style* não seja none.

0.8 Exercício

Aproveitando o exerício da aula 1, agora iremos melhorar a formatação. O objetivo deste exercício é aplicar os conceitos acima na formatação da página. Desta vez, será necessário seguir o modelo wireframe como fora definido. Lembrando, organizem seu projeto como descrito em 0.4.