**Motivação**

HTML (abreviação de [Hypertext Markup Language](http://www.devmedia.com.br/guias/html/3)) é uma linguagem de marcação utilizada na estruturação de páginas web. Sua sintaxe é bastante simples e, assim como a XML é baseada em tags, que representam os diversos elementos de uma página, como imagens e links.

Ao acessar uma página web através de um navegador, ele é capaz de interpretar o código HTML e renderizá-lo de forma compreensível para o usuário final, exibindo textos, botões, etc. com as configurações definidas por meio das diversas tags que essa linguagem dispõe.

Atualmente a **HTML** encontra-se na versão 5 e é padronizada pelo W3C (World Wide Web Consortium), uma organização internacional responsável por estabelecer padrões para a internet, como a linguagem XML, as CSS e o SOAP.

**Estrutura básica de uma página HTML**

A estrutura básica de uma [página HTML](http://www.devmedia.com.br/front-end-web/) pode ser vista na **Listagem 1**, na qual podemos ver as principais tags que são necessárias para que o documento seja corretamente interpretado pelos browsers.

01 <!DOCTYPE html>

02 <html>

03 <head>

04 <meta charset="UTF-8"/>

05 <title>Document</title>

06 </head>

07 <body>

08 <!-- Conteúdo -->

09 </body>

10 </html>

**Listagem 1**. Estrutura básica de uma página HTML

Linha 1: a instrução DOCTYPE deve ser sempre a primeira a aparecer em uma página HTML e indica para o browser qual versão da linguagem está sendo usada. Nesse caso, estamos trabalhando com a HTML5, versão na qual a declaração do DOCTYPE é bastante simples, como podemos ver na listagem;

Linhas 2 e 10: abertura e fechamento da tag html, que delimita o documento. Sendo assim, todas as demais tags da página devem estar nesse espaço;

Linhas 3 e 6: abertura e fechamento da tag head, que define o cabeçalho do documento. O conteúdo nesse espaço não é visível no browser, mas contém instruções sobre seu conteúdo e comportamento. Dentro dessa tag, por exemplo, podem ser inseridas folhas de estilo e scripts;

Linha 4: a tag meta, nesse caso, especifica qual conjunto de caracteres (character set ou charset) será usado para renderizar o texto da página. O UTF-8 contém todos os caracteres dos padrões Unicode e ASCII, sendo, portanto, o mais utilizado em páginas web. A mesma tag meta, porém com outros atributos, pode ser utilizada para outros fins, como na SEO (Search Engine Optimization);

Linha 5: a tag title define o título da página, aquele que aparece na janela/aba do navegador;

Linhas 7 e 9: abertura e fechamento da tag body, marcando o espaço no qual deve estar contido o conteúdo visual da página. As demais tags que representam texto, botões, etc. devem ser adicionadas nesse intervalo;

Linha 8: nessa linha podemos observar a sintaxe para adição de comentários em HTML. Esse trecho não é renderizado pelo browser.

Para desenvolver páginas com HTML, basicamente precisamos de um editor de texto, como o Bloco de Notas do Windows, Nano e Emacs no Linux, entre vários outros. Há, ainda, editores com opções avançadas, como recursos de syntax hilghligt e autocomplete, como Sublime Text, Atom, Brackets e Visual Studio Code, que podem ser usados para editar documentos HTML.

Independente do editor utilizado, podemos simplesmente copiar o código da **Listagem 1** para um deles e salvar o arquivo com extensão .html. Em seguida, podemos abrir esse arquivo em um browser.

**Tags**

Um documento HTML é composto por tags, as quais possuem um nome e aparecem entre os sinais < e >, como vimos na **Listagem 1**, por exemplo, em <html> e <head>. Naquele exemplo também vimos que algumas tags precisam ser abertas e fechadas, como em <body></body>. Nesse caso, a tag de fechamento deve conter a barra / antes do nome. Outras, porém, não precisam ser fechadas, como a tag <meta>. Nesses casos, a adição da barra / no final da própria tag, como vemos na linha 4 da **Listagem 1**, é opcional.

Outro aspecto importante da linguagem é que ela é case insensitive, ou seja, não leva em consideração a diferença entre letras maiúsculas e minúsculas. No entanto, o uso apenas de letras minúsculas tem sido utilizado como padrão pelos desenvolvedores.

Além dessas características, as tags também possuem atributos, como vemos na linha 4 da **Listagem 1**, na qual a tag meta possui o atributo charset=”UTF-8”. Essas propriedades definem algumas características adicionais de cada tag e em alguns casos são obrigatórias. Seus valores devem aparecer entre aspas duplas, como no exemplo acima, ou em aspas simples, caso o próprio valor contenha aspas duplas.

**Cabeçalhos**

Cabeçalhos são normalmente utilizados para identificar páginas e seções, e possuem aparência diferenciada do restante do texto. No HTML há seis níveis de cabeçalhos/títulos que podem ser utilizados por meio das tags h1, h2, h3, h4, h5 e h6, sendo h1 o maior/mais relevante e h6 o menor/menos relevante.

Na **Listagem 2** podemos ver como utilizar essas tags, e na Figura N temos o resultado desse código. Para testar esse código, basta copiá-lo para o espaço indicado na linha 8 da **Listagem 1** e salvar o documento com a extensão .html.

<h1>Título de nível 1</h1>

<h2>Título de nível 2</h2>

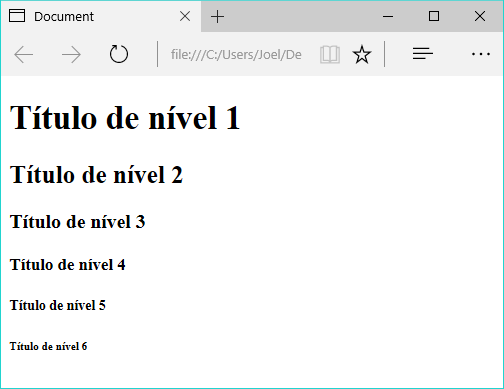
<h3>Título de nível 3</h3>

<h4>Título de nível 4</h4>

<h5>Título de nível 5</h5>

<h6>Título de nível 6</h6>

**Listagem 2**. Diferentes níveis de cabeçalho

  
**Figura 1.** Diferentes níveis de cabeçalho

**Parágrafos**

Parágrafos de texto são gerados na HTML por meio das tags

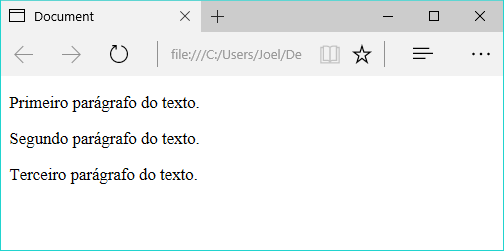
. Esse é um exemplo de tag cuja disposição na tela se dá em forma de bloco, ou seja, um parágrafo é posto sempre abaixo do outro.

O código a seguir adiciona três parágrafos, que podem ser visualizados na Figura 2.

<p>Primeiro parágrafo do texto.</p>

<p>Segundo parágrafo do texto.</p>

<p>Terceiro parágrafo do texto.</p>

  
**Figura 2.** Parágrafos visualizados na página

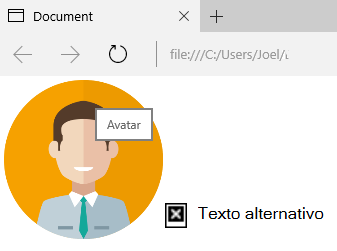
**Imagens**

A inserção de imagens em uma **página HTML** pode ser feita por meio da tag img, que recebe no atributo src o endereço do arquivo a ser carregado. Além desse, outros dois atributos importantes são o alt, que indica um texto alternativo que será exibido caso o arquivo não possa ser carregado, e title, que indica o texto que aparecerá como tooltip ao passar o mouse sobre a figura.

O código a seguir insere uma imagem a partir do arquivo perfil.png, localizado na mesma pasta do arquivo HTML, e uma tag img apontando para um arquivo inexistente. Observe na Figura 3 que enquanto a primeira aparece corretamente, a segunda é exibida com um ícone de erro o texto alternativo que informamos. Note, ainda, que sobre a primeira figura está sendo exibido o tooltip definido no atributo title.

<img src="avatar.png" alt="Texto alternativo" title="Avatar" />

<img src="arquivo\_inexistente.jpg" alt="Texto alternativo" title="Outra figura"/>

  
**Figura 3.** Exemplos de uso da tag img

**Links**

Links são normalmente utilizados para direcionar o usuário para outras páginas, ou para outras partes da mesma página. Nos dois casos, utilizamos a tag a, que possui o atributo href no qual indicamos o destino daquele link.

O seguinte código mostra como adicionar um link para outra página, neste caso, indicada pelo arquivo pagina2.html.

<a href="pagina2.html">Página 2</a>

Já o código abaixo mostra como adicionar um link para um elemento na mesma página. Nesse caso, ao clicar no link o browser mudará o foco para o elemento que possui o atributo id igual àquele indicado no href.

<a href="#paragrafo3">Ir para o parágrafo 3</a>

<!--outros elementos-->

<p id="rodape">Parágrafo no rodapé.</p

Note também que nesse caso o href requer ainda o sinal de cerquilha (#) antes do id do elemento que será o foco do link.

**Tabelas**

Tabelas são elementos utilizados com frequência para exibir dados de forma organizada em linhas e colunas. No HTML, elas são formadas por três tags básicas: table, para delimitar a tabela; tr, para indicar as linhas; e td para formar as colunas. A **Listagem 3** mostra um exemplo simples de tabela com 3 linhas e duas colunas, e seu resultado pode ser visto na Figura 4.

<table>

<tr>

<td>Linha 1, Coluna 1</td>

<td>Linha 1, Coluna 2</td>

</tr>

<tr>

<td>Linha 2, Coluna 1</td>

<td>Linha 2, Coluna 2</td>

</tr>

<tr>

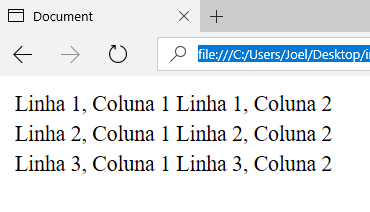
<td>Linha 3, Coluna 1</td>

<td>Linha 3, Coluna 2</td>

</tr>

</table>

**Listagem 3**. Exemplo de uso de tabelas

  
**Figura 4.** Exemplo de tabela com 3 linhas e 2

Existem ainda outras três tags utilizadas para delimitar, de forma mais organizada, as partes da tabela: thead para o cabeçalho; tbody para o corpo; e tfoot para o rodapé. A **Listagem 4** traz um exemplo de tabela mais complexa, utilizando todas as tags. Seu resultado é apresentado na Figura 5.

<table>

<thead>

<tr>

<th>Produto</th>

<th>Preço</th>

<th>Quantidade</th>

</tr>

</thead>

<tbody>

<tr>

<td>Feijão</td>

<td>R$ 8,75</td>

<td>1</td>

</tr>

<tr>

<td>Arroz</td>

<td>R$ 4,99</td>

<td>2</td>

</tr>

</tbody>

<tfoot>

<tr>

<td>Total</td>

<td></td>

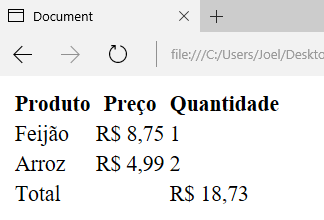
<td>R$ 18,73</td>

</tr>

</tfoot>

</table>

**Listagem 4**. Tabela com cabeçalho e rodapé

  
**Figura 5.** Tabela com cabeçalho e rodapé

*Observação: Por padrão, as tabelas não possuem bordas. Isso deve ser adicionado por meio das CSS.*

**Listas**

Listas são elementos úteis para organizar e ordenar itens que estão relacionados de alguma forma. No HTML é possível criar três tipos de listas: ordenadas (com a tag ol), não ordenadas (com a tag ul), e de definição (por meio da tag dl).

Na **Listagem 5** temos um exemplo de lista ordenada, enquanto a **Listagem 6** traz um exemplo de lista não ordenada. Em seguida, a Figura 6 mostra o resultado dos dois códigos. Observe que cada item das primeiras listas é definido pela tag li.

<ol>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ol>

**Listagem 5**. Lista ordenada

<ul>

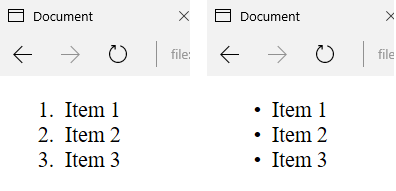
<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

**Listagem 6**. Lista não ordenada

  
**Figura 6.** Listas ordenada e não ordenada

As listas de definição têm um comportamento um pouco diferente, uma vez que cada item é composto por um título (dt) e uma definição (dd), semelhante ao que ocorre em dicionários, nos quais temos os verbetes e suas definições. Na **Listagem 7** temos um exemplo desse tipo de lista, com seu resultado ilustrado na Figura 7.

<dl>

<dt>Título 1</dt>

<dd>Definição 1</dd>

<dt>Título 2</dt>

<dd>Definição 2</dd>

<dt>Título 3</dt>

<dd>Definição 3</dd>

</dl>

**Listagem 7**. Exemplo de lista de definição

**Formulários**

Formulários são normalmente utilizados para integrar a página HTML a algum processamento no lado servidor. Nesses casos, a página envia dados para uma aplicação (Java, PHP, .NET, etc), que os recebe, trata e retorna algum resultado.

No HTML, geralmente usamos a tag form para delimitar a área na qual se encontram os campos a serem preenchidos pelo usuário, a fim de serem enviados para processamento no back-end (enquanto a página HTML é chamada de front-end da aplicação).

A **Listagem 8** mostra um exemplo de formulário com vários tipos de campos para entrada de dados e um botão para submetê-los ao servidor. Na Figura 8 vemos seu resultado.

<form action="salvar\_dados.php" method="post">

<input type="text" placeholder="Digite aqui"> <br/>

<input type="checkbox">Checkbox <br/>

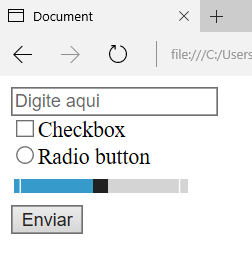
<input type="radio">Radio button <br/>

<input type="range"> <br/>

<input type="submit" value="Enviar">

</form>

**Listagem 8**. Formulário com campos

  
**Figura 8.** Formulário para envio de dados

**Formatação de texto**

As tags de formatação de texto ajudam a destacar trechos da parte escrita da página, seja para fins de SEO ou por requisitos do conteúdo. Formatações como negrito e itálico podem ser aplicadas com facilidade utilizando as várias tags disponíveis para esse fim:

* **b** e **strong** para negrito/texto forte;
* **i** e **em** para itálico/ênfase;
* **sup** e **sub** para sobrescrito e subscrito, respectivamente;
* **ins** e **del** para indicar trechos que foram incluídos ou removidos, respectivamente;
* **small** para textos menores que o padrão;
* **mark** para texto destacado.

Na **Listagem 9** vemos exemplos de uso dessas tags, e seu resultado pode ser observado em seguida, na Figura 9.

<p>Texto em negrito com <b>bold</b> e <strong>strong</strong>.</p>

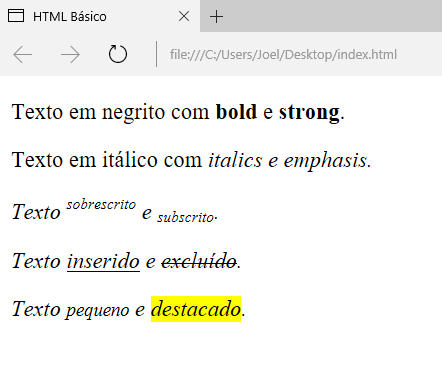
<p>Texto em itálico com <i>italics<i> e <em>emphasis</em>.</p>

<p>Texto <sup>sobrescrito</sup> e <sub>subscrito</sub>.</p>

<p>Texto <ins>inserido</ins> e <del>excluído</del>.</p>

<p>Texto <small>pequeno</small> e <mark>destacado</mark>.</p>

**Listagem 9**. Tags de formatação de texto

  
**Figura 9.** Textos com formatação especial

**Áudio**

Com HTML também é possível inserir áudio e vídeo nas páginas com facilidade. Para áudio, podemos utilizar a tag **audio**, da seguinte forma:

<audio controls src="musica.mp3" >

Seu browser não suporta áudio.

</audio>

Na tag audio, o atributo **src** aponta para o arquivo de áudio que será executado (MP3, OGG ou WAV). Já o atributo **controls** indica que devem ser exibidos os controles de gerenciamento do áudio (botões play, pause, etc). Além dele, outros também merecem destaque: **autoplay**, para fazer com que o áudio seja executado assim que for carregado; **loop**, para que o áudio seja executado repetidas vezes.

*Observação: Caso o browser não ofereça suporte a essa tag, o texto contido eu seu interior será exibido.*

Também podemos informar mais de um arquivo de áudio (opções alternativas de formato, por exemplo). Nesse caso, precisamos utilizar a tag **source**, como no exemplo a seguir:

<audio controls src="musica.mp3" >

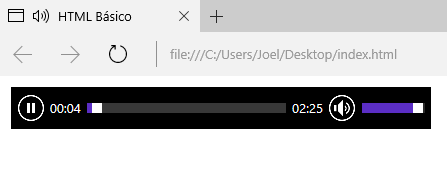
<source src="musica.ogg" type="audio/ogg">

<source src="musica.mp3" type="audio/mpeg">

</audio>

Dessa vez, os arquivos serão buscados na ordem em que aparecem na lista. Caso um não seja localizado ou não possa ser carregado, o browser buscará imediatamente o próximo.

A **Figura 10** ilustra o resultado desses códigos, com o controle de áudio exibido no browser. Note também que no topo da janela (na aba do navegador) é exibido um ícone informando ao usuário que essa página está executando um áudio.

  
**Figura 10.** Áudio em execução na página

**Vídeo**

Semelhante ao áudio, também podemos inserir vídeos nas páginas HTML utilizando a tag **vídeo**, adicionada na HTML5, da seguinte forma:

<video src="video.mp4" width="640" height="480" controls>

Seu browser não suporta vídeo HTML5.

</video>

Dessa vez, além dos atributos já vistos no áudio, também precisamos informar a largura (**width**) e a altura (**height**) do vídeo, a fim de mantê-lo adequado ao layout.

O código a seguir pode ser utilizado quando desejarmos informar mais de um arquivo de vídeo como opções de carregamento, da mesma forma que vimos na tag **audio**. Nesse caso, os formatos suportados são MP4, WebM e Ogg.

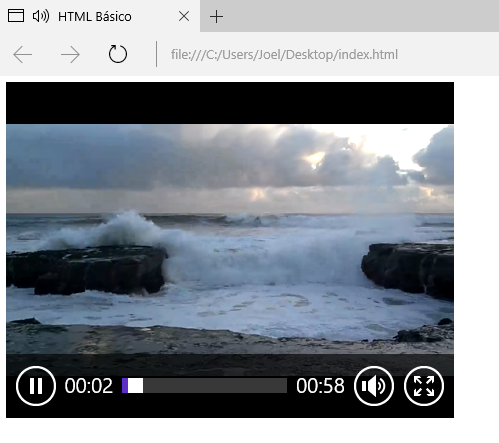
<video width="640" height="480" controls>

<source src="vdeio.webm" type="video/webm">

<source src="video.mp4" type="video/mp4">

</video>

O resultado agora pode ser visto na Figura 11.

  
**Figura 11.** Vídeo em execução na página

**Div e Span**

As [tags div e span](http://www.devmedia.com.br/como-centralizar-divs-em-html-e-css/37568) são duas das mais utilizadas no HTML, com objetivos distintos, porém com grande importância para a composição do layout das páginas e formatação do texto.

As divs são normalmente utilizados para representarem containers para outros elementos, agrupando-os visualmente dentro de um bloco que pode conter dimensões e posição definidas. Por padrão, uma div não possui aparência características visuais definidas, isso precisa ser feito via CSS ao atribuir bordas, cores, etc. Sua principal característica, no entanto, é que essa tag representa um elemento do tipo bloco, ou seja, que quando adicionado na página, automaticamente gera uma nova linha no layout (semelhante a um parágrafo), ao invés de ser alocado lateralmente nos demais componentes.

O código a seguir demonstra um uso básico das divs:

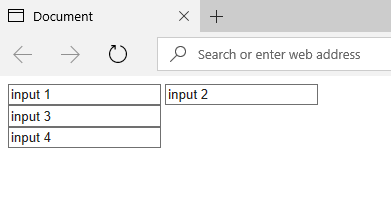
<input type="text" value="input 1">

<input type="text" value="input 2">

<div><input type="text" value="input 3"></div>

<div><input type="text" value="input 4"></div>

Perceba no resultado da **Figura 12** que, enquanto os dois primeiros inputs são dispostos lateralmente, os dois últimos aparecem um abaixo do outro, uma vez que estão dentro de divs diferentes.

  
**Figura 12.** Exemplos de uso das divs

Já a tag span é um elemento do tipo inline, ou seja, quando adicionado na página, ele é inserido lateralmente após os demais componentes, diferente das divs que são elementos do tipo bloco.

Elementos span, por padrão, também não possuem nenhuma característica visual definida, isso precisa ser feito via CSS para destacar ou aplicar uma formatação especial para um certo trecho do texto. Por exemplo, o código abaixo demonstra o uso do span em dois casos. No primeiro, a tag não conta com nenhum atributo adicional; no segundo, adicionamos a ela uma aparência diferenciada via CSS.

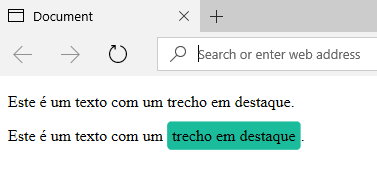
<p>Este é um texto com um <span>trecho em destaque</span>.</p>

<p>

Este é um texto com um <span style="background: #1abc9c; border-radius:5px; padding:5px">trecho em destaque</span>.

</p>

O resultado pode ser visto na **Figura 13** e, como esperado, no primeiro caso não conseguimos perceber nenhuma diferença visual devido ao uso do span.

  
**Figura 13.** Exemplo de uso do span