

AULÃO DESIGN PARA WEB

Professora: Joseneuza Aguiar

Aluno: Lucas Barbosa

HTML

Tags Importantes 1:

`<!DOCTYPE html>`

Durante os anos de desenvolvimento do HTML, os padrões da web ainda não foram acordados entre todas as partes. Os fornecedores de navegadores criariam novos recursos da maneira que quisessem. Havia pouca preocupação com navegadores concorrentes.

O resultado foi que os desenvolvedores web tiveram que escolher um navegador para desenvolver seus sites. Isso significava que os sites não renderizariam bem em navegadores não suportados, o que é uma situação insustentável.

O W3C (World Wide Web Consortium) escreveu um conjunto de padrões web para lidar com essa situação. Todos os fornecedores de navegadores e desenvolvedores da Web devem aderir a esses padrões. Isso garantiria que os sites renderiam bem em todos os navegadores.

As mudanças exigidas pelas normas foram bem diferentes de algumas práticas existentes. Aderir a eles quebraria sites existentes não compatíveis com os padrões.

Para lidar com esse problema, os fornecedores começaram a programar modos de renderização em seus navegadores. Os desenvolvedores da Web precisariam adicionar uma declaração de DOCTYPE no início de um documento HTML. A declaração DOCTYPE informaria ao navegador qual modo de renderização usar para esse documento.

Três modos de renderização separados estavam geralmente disponíveis nos navegadores.

Modo de padrões completo - Renderiza as páginas de acordo com os padrões da web W3C.

Modo Quirks - Renderiza páginas de maneira não compatível com os padrões.

Modo quase padrão - É próximo do modo padrão completo, mas oferece suporte para um pequeno número de peculiaridades.

Na era moderna do HTML5, os padrões da Web são totalmente implementados em todos os principais navegadores. Os sites da Web geralmente são desenvolvidos de maneira compatível com os padrões. Por causa disso, a declaração do tipo de documento HTML5 existe apenas para informar ao navegador para renderizar o documento no modo de padrões completos.

Conclusão: Desde o HTML5, o DOCTYPE não é um requisito obrigatório. Porém, a presença do DTD garante que o navegador se comportará adequadamente e de maneira precisa.

`<html>`

A tag **<html>** representa a raiz de um documento HTML e é o contêiner para todos os outros elementos HTML, com exceção do DOCTYPE.

<head>

O elemento **<head>** é um contêiner para metadados (dados sobre dados) e é colocado entre a tag **<html>** e a tag **<body>**. Metadados são dados sobre o documento HTML. Os metadados não são exibidos para o usuário. Metadados normalmente definem o título do documento, conjunto de caracteres, estilos, scripts, etc.

<body>

A tag **<body>** define o corpo do documento. O elemento **<body>** contém todo o conteúdo de um documento HTML, como títulos, parágrafos, imagens, hiperlinks, tabelas, listas, etc.

<!--...-->

A tag comment (**<!-- -->**) é usada para inserir comentários no código-fonte. Os comentários não são exibidos nos navegadores, porém podem ser inspecionados via navegador. Você pode usar comentários para explicar seu código, o que pode ajudá-lo a editar o código-fonte posteriormente. Útil se você tiver um código bem grande.

Tags Importantes 2 - head:

<title></title>

Todo documento em HTML deve possuir um título. O título aparece em lugar separado da página, e é utilizado para identificar o documento em outros contextos (por exemplo, buscas, SEO, etc). É interessante que o título possa sugerir claramente o conteúdo do documento.

Atenção porque o conceito de título é diferente de cabeçalho. O título está mais para o nome do arquivo. Não é um elemento relevante na visualização do documento como acontece com o cabeçalho.

<style></style>

É usada para definir informações de estilo (CSS) para um documento. Dentro do elemento **<style>** você especifica como os elementos HTML devem ser renderizados em um navegador. É o tal do CSS interno.

<script></script>

A tag **<script>** é usada para incorporar um script do lado do cliente (JavaScript). O elemento **<script>** contém instruções de script ou aponta para um arquivo de script externo por meio do atributo src. Este já é o JavaScript interno.

<meta>

A tag **<meta>** define metadados sobre um documento HTML. Metadados são dados (informações) sobre dados. Tags **<meta>** sempre vão dentro do elemento **<head>** e são normalmente usadas para especificar conjunto de caracteres, descrição de página, palavras-chave, autor do documento e configurações da janela de visualização.

Os metadados não serão exibidos na página, mas é analisável por máquina. Os metadados são usados por navegadores (como exibir conteúdo ou recarregar a página), mecanismos de pesquisa (palavras-chave) e outros serviços da web.

Tags Importantes 3 - body:

Tag Break Row

<p> Parágrafo Um </p>

<p> Parágrafo Dois </p>

A tag
 insere uma única quebra de linha. É uma tag vazia, o que significa que não possui tag final.

Tag Paragraph

<p> “Às vezes, a vida vai te acertar um tijolo na cabeça. Não perca a fé.

Eu estou convencido de que a única coisa que me fez seguir em frente

era que eu amava o que fazia.” </p>

O elemento HTML <p> representa um parágrafo. Em mídias visuais, parágrafos são representados como blocos indentados de texto com a primeira letra avançada e separados por linhas em branco. Já em HTML, parágrafos são usados para agrupar conteúdos relacionados de qualquer tipo, como imagens e campos de um formulário.

Tag Heading

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

As tags <h1> a <h6> são usadas para definir cabeçalhos HTML. <h1> define o título mais importante. <h6> define o título menos importante.

Tag Image

A tag é usada para incorporar uma imagem em uma página HTML. As imagens não são tecnicamente inseridas em uma página da web; as imagens estão vinculadas a páginas da web. A tag cria um espaço de retenção para a imagem referenciada.

A tag tem dois atributos obrigatórios:

src - Especifica o caminho para a imagem

alt - Especifica um texto alternativo para a imagem, se a imagem por algum motivo não puder ser exibida

Tag IFrame

<iframe width="560" height="315"
src="https://www.youtube.com/embed/oqKfkIpUQI4" frameborder="0"
allow="accelerometer; autoplay; encrypted media; gyroscope; picture-in-picture"
allowfullscreen></iframe>

O elemento HTML **<iframe>** (ou elemento HTML inline frame) representa um contexto de navegação aninhado, efetivamente incorporando outra página HTML para a página atual. Um documento pode conter uma cabeça e um corpo ou uma cabeça e um conjunto de quadros, mas não tanto um corpo e um conjunto de quadros. No entanto, um **<iframe>** pode ser usado dentro de um corpo de documento normal. Cada contexto de navegação tem sua própria história de sessão e o documento ativo. Um quadro embutido(iframe) é usado para incorporar outro documento no documento HTML.

Tag Video

```
<video width =" height =" controls  
    <source src="coringa.mp4" type="video/mp4">  
    <source src="coringa.ogg" type="video/ogg">  
    Não foi possível carregar o vídeo!  
</video>
```

A tag **<video>** é usada para incorporar vídeo em um documento HTML, como um clipe de filme ou outros fluxos de vídeo. A tag **<video>** contém uma ou mais tags **<source>** com diferentes origens de vídeo. O navegador escolherá a primeira fonte que ele suporta.

O texto entre as tags **<video>** e **</video>** só será exibido em navegadores que não suportam o elemento **<video>**.

Tag Anchor

```
<a href="https://www.google.com/">Pesquisa no Google</a>  
<a href="steve.html">Steve Jobs</a>  
<a href="#texto_id">Texto_Id</a>
```

A tag **<a>** define um hyperlink, que é usado para vincular uma página a outra. O atributo mais importante do elemento **<a>** é o atributo **href**, que indica o destino do link.

Tag Ordered List

```
<ol>  
    <li>Banana</li>  
    <li>Laranja</li>  
    <li>Maça</li>  
    <li>Uva</li>  
</ol>
```

A tag **** define uma lista de itens ordenados. forma característica esses itens ordenados em uma lista são mostrados com uma contagem que os precede, que pode ser de qualquer tipo, como numerais, letras, algarismos romanos, ou simples símbolos.

Tag Unordered List

```
<ul>  
    <li>Banana</li>
```

```
<li>Laranja</li>
<li>Maça</li>
<li>Uva</li>

</ul>
```

O elemento HTML **** (ou elemento HTML de Lista desordenada) representa uma lista de itens sem ordem rígida, isto é, uma coleção de itens que não trazem uma ordenação numérica e as suas posições, nessa lista, são irrelevantes. Caracteristicamente, os itens em uma lista desordenada são exibidos com um marcador que pode ter várias formas, como um ponto, um círculo, ou um quadrado.

Tabelas em HTML

<table> → tabela

<tr> → linha da tabela

<th> → cabeçalho da tabela

<td> → célula da tabela

```
<table>
  <tr>
    <th>Nome</th>
    <th>Cargo</th>
    <th>Departamento</th>
  </tr>
  <tr>
    <td>João</td>
    <td>Gerente</td>
    <td>Vendas</td>
  </tr>
  <tr>
    <td>Maria</td>
    <td>Aux. Administrativo</td>
    <td>Administração</td>
  </tr>
</table>
```

<caption> → título da tabela

```
<table>
  <caption>Poupança</caption>
```

```

<tr>
  <th>Mês</th>
  <th>Dinheiro guardado</th>
</tr>
<tr>
  <td>janeiro</td>
  <td>$100</td>
</tr>
<tr>
  <td>fevereiro</td>
  <td>$200</td>
</tr>
</table>

```

<thead> → agrupar o conteúdo do cabeçalho em uma tabela HTML.

<tbody> → agrupar o conteúdo do corpo em uma tabela HTML.

<tfooter> → agrupar o conteúdo do rodapé em uma tabela HTML.

<thead>, <tbody>, <tfooter> → Os navegadores podem usar esses elementos para permitir a rolagem do corpo da tabela independentemente do cabeçalho e do rodapé. Além disso, ao imprimir uma tabela grande que abrange várias páginas, esses elementos podem permitir que o cabeçalho e o rodapé da tabela sejam impressos na parte superior e inferior de cada página.

```

<table>
  <thead>
    <tr>
      <th>Poupança</th>
      <th>Mensal</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>janeiro</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>fevereiro</td>

```

```

        <td>$80</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <td>Soma</td>
        <td>$180</td>
    </tr>
</tfoot>
</table>

```

`<colgroup>` → Especifica um grupo de uma ou mais colunas em uma tabela para formatação. É útil para aplicar estilos a colunas inteiras, em vez de repetir os estilos para cada célula, para cada linha.

```

<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>
<h1>The colgroup element</h1>
<table>
    <colgroup>
        <col span="2" style="background-color:red">
        <col style="background-color:yellow">
    </colgroup>
    <tr>
        <th>ISBN</th>
        <th>Title</th>
        <th>Price</th>
    </tr>
    <tr>

```

```

        <td>3476896</td>
        <td>My first HTML</td>
        <td>$53</td>
    </tr>
    <tr>
        <td>5869207</td>
        <td>My first CSS</td>
        <td>$49</td>
    </tr>
</table>
</body>
</html>

```

Formulários em HTML

<form> → form

<caption> → título do form

<input> → entrada de dados

<label> → identificador da entrada de dados

<textarea> → grande campo de texto

<fieldset> → separar inputs html em grupos diferentes

<legend> → Tag associada à fieldset. Usada para nomear o grupo

<select> → Lista com várias opções.

<option> → Tag associada à select. Contém as opções

HTML Semântico

<header> → *tag* específica para o cabeçalho da página. auxiliando para casos em que você possui uma *logo* ou nome da página em destaque, ou até mesmo título de artigo.

<aside> → Indica conteúdos laterais, geralmente utilizada para menus laterais. Essa *tag* nos ajuda a definir, até mesmo para a própria compreensão do código, os conteúdos tratados nas laterais da página.

<section> → que define seções de conteúdos — seja dentro de um texto ou conteúdo principal, conseguimos dividir melhor esse conteúdo, deixando-o melhor organizado.

Daí você pode me perguntar: “Qual seria a diferença de uma <section> para uma <div>?”

Pois bem, as <section> conseguem separar os conteúdos de forma diferente das <div>, já que para utilizar um título <h1 a h6> em uma <div> **não podemos usar o mesmo título para várias <div>.**

Isso ocorre por que, ao passar pela pesquisa do google, algumas informações da página na busca poderiam se perder.

Já com a `<section>` podemos utilizar o mesmo título, pois as `<section>` separam seus conteúdos por seções. Portanto, quando pesquisado em mecanismos de busca, não se perdem informações.

`<article>` → é um artigo dentro do conteúdo. Pode ser uma citação de textos externos, ou referências, geralmente também usamos um **título** para essa tag.

`<nav>` → usamos para **menus interativos**, como por exemplo menus que encaminham a página de “sobre” ou “produtos”. O `<nav>` pode ser usado em qualquer momento na página, onde tenhamos **agrupamento de links**.

`<main>` → tag onde colocamos o conteúdo principal da página. Para melhor uso, só devemos ter um `<main>` por página, onde iremos apontar o conteúdo de maior relevância, sendo textos, imagens, links, etc.

Essa *tag* facilitou muito a compreensão para leitores de tela, já que **ênfatiza o texto dentro da tag**, deixando clara sua importância.

`<figure>` → é utilizada para a inserção de figuras na página. Juntamente com essa tag trabalhamos com o *alt*, onde descrevemos o que há na imagem, facilitando então a acessibilidade da página, para os leitores de tela.

Além disso, se por algum motivo a imagem não abrir, haverá uma descrição dela para quem estiver acessando saber sobre o que se trata.

`<footer>` → é a tag utilizada para o rodapé da página ou até de rodapé para artigos.

Normalmente utilizada para inserir a autoria da página, assim como contatos e data de criação de conteúdo.

CSS

Considerações Importantes:

É padrão de mercado e boa prática separar os atributos css em um documento externo .css e vinculá-lo via <link rel="stylesheet" href="seuEstilo.css" type="text/css">

Três maneiras de inserir css (Externo, interno e embutido na tag)

O DOM (Document Object Model) é a representação de dados dos objetos que compõem a estrutura e o conteúdo de um documento na Web.

Todo o DOM é manipulado no CSS via seletores fazendo referência ou à própria tag, ou à um id, ou à uma classe.

A grande maioria das tag podem possuir o atributo "class" ou "id"

```
h1 {  
    color: #00c300;  
}  
.primeiroHeading {  
    color: #01c301;  
}  
#primeiroHeadingPagina {  
    color: #9fc39f;  
}  
h1, h2 {  
    color: #aa002f;  
}  
.segundoHeading, .terceiroHeading {  
    color: #9b5a0f;  
}  
#paragrafoEmpresa, #paragrafoFornecedor {  
    color: #8aa02f;  
}
```

Pseudo Classes

| Pseudo Classe | Exemplo | Descrição |
|-----------------------------------------------|-----------------|-------------------------------------------------------------------------------|
| <u>:active</u> | a:active | Seleciona o link ativo |
| <u>:checked</u> | input:checked | Seleciona cada elemento <input> verificado |
| <u>:disabled</u> | input:disabled | Seleciona cada elemento <input> desativado |
| <u>:empty</u> | p:empty | Seleciona cada elemento <p> que não tem filhos |
| <u>:enabled</u> | input:enabled | Seleciona cada elemento <input> ativado |
| <u>:first-child</u> | p:first-child | Seleciona todos os elementos <p> que é o primeiro filho de seu pai |
| <u>:first-of-type</u> | p:first-of-type | Seleciona cada elemento <p> que é o primeiro elemento <p> de seu pai |
| <u>:focus</u> | input:focus | Seleciona o elemento <input> que tem foco |
| <u>:hover</u> | a:hover | Seleciona links no mouse |
| <u>:in-range</u> | input:in-range | Seleciona elementos <input> com um valor dentro de uma faixa especificada |
| <u>:invalid</u> | input:invalid | Seleciona todos os elementos <input> com um valor inválido |
| <u>:lang(<i>language</i>)</u> | p:lang(it) | Seleciona cada elemento <p> com um valor de atributo lang começando com "ele" |
| <u>:last-child</u> | p:last-child | Seleciona cada <p> elementos que é o último filho de seu pai |

| | | |
|---------------------------------------------|-----------------------|--------------------------------------------------------------------------------------------------------|
| <u>:last-of-type</u> | p:last-of-type | Seleciona cada elemento <p> que é o último elemento <p> de seu pai |
| <u>:link</u> | a:link | Seleciona todos os links não visitados |
| <u>:not(selector)</u> | :not(p) | Seleciona cada elemento que não é um elemento <p> |
| <u>:nth-child(n)</u> | p:nth-child(2) | Seleciona cada elemento <p> que é o segundo filho de seu pai |
| <u>:nth-last-child(n)</u> | p:nth-last-child(2) | Seleciona cada elemento <p> que é o segundo filho de seu pai, contando desde o último filho |
| <u>:nth-last-of-type(n)</u> | p:nth-last-of-type(2) | Seleciona cada elemento <p> que é o segundo elemento <p> de seu pai, contando a partir do último filho |
| <u>:nth-of-type(n)</u> | p:nth-of-type(2) | Seleciona cada elemento <p> que é o segundo elemento <p> de seu pai |
| <u>:only-of-type</u> | p:only-of-type | Seleciona cada elemento <p> que é o único elemento <p> de seu pai |
| <u>:only-child</u> | p:only-child | Seleciona cada elemento <p> único filho de seu pai |
| <u>:optional</u> | input:optional | Seleciona elementos <input> sem atributo "necessário" |
| <u>:out-of-range</u> | input:out-of-range | Seleciona elementos <input> com um valor fora de uma faixa especificada |
| <u>:read-only</u> | input:read-only | Seleciona <input> elementos com um atributo "readonly" especificado |

| | | |
|------------------------------------|------------------|------------------------------------------------------------------------------------------|
| <u>:read-write</u> | input:read-write | Seleciona <input> elementos sem atributo "readonly" |
| <u>:required</u> | input:required | Seleciona <input> elementos com um atributo "obrigatório" especificado |
| <u>:root</u> | root | Seleciona o elemento raiz do documento |
| <u>:target</u> | #news:target | Seleciona o elemento ativo atual #news (clicado em uma URL contendo esse nome de âncora) |
| <u>:valid</u> | input:valid | Seleciona todos os elementos <input> com um valor válido |
| <u>:visited</u> | a:visited | Seleciona todos os links visitados |

Atributos importantes:

Globais/São de muitas tags → margin(Recebe 1, 2, 3 ou 4 parametros); margin-top; margin-bottom; margin-right; margin-left;

border;

padding(Recebe 1, 2, 3 ou 4 parametros); padding-top; padding-bottom; padding-left; padding-right;

Tags textuais(h(1,2,3,4,5,6), p,span,strong,pre,div) → color; background-color;

text-align; text-align-last;

text-decoration-line; text-decoration-color; text-decoration-style; text-decoration-thickness; text-decoration;

text-transform;

text-indent; letter-spacing; word-spacing; line-height;

text-shadow;

font(Recebe 4 parametros); font-style; font-weight; font-family; font-size; font-variant;

body -> background-color(nomes em inglês, hexadecimal ou rgb()); background-image(link na web ou dentro do próprio diretório); background-color(nomes em inglês, hexadecimal ou rgb());

ol → type("1","A","a","I","i");

ul → type("disc","circle","square","none");

table → border;

th → rowspan; colspan;

td → rowspan; colspan;

form → name; method; action;

input → type; name; id; size; maxlength; required;

textarea → rows; cols;

Cores:

Para representar o vermelho –R:255 –G:0 –B: 0

Para representar o verde –R:0 –G:255 –B: 0

Para representar o azul –R:0 –G:0 –B: 255

Hexadecimal:

Branco: #FFFFFF

Preto: #000000

RGB:

Branco: rgb(255,255,255)

Preto: rgb(0,0,0)

BOOTSTRAP

Considerações Importantes:

É um framework CSS. Tem estilizações pré-prontas. Para importar, pode ser via link (no momento de execução do arquivo, deve-se ter internet) ou arquivo local baixado do site (recomendado).

Componentes Interessantes:

1. Caixas expansivas (sanfona e classe de colapso) - *Accordion*
2. Caixa de alertas (*alert*) e caixa de notificações (*toast*)
3. Emblemas - *Badge*
4. Botões (*Button*), grupo de botões(*Button group*) e *dropdown* -
5. *Cards*
6. Barras de progresso - *Progress*
7. Carrossel de imagens - *Carousel*
8. Caixas de diálogos - *Modal*
9. *Popovers* e *tooltips*
10. Menus: barras de navegação e abas - *Nav*

Fontes:

Materiais da profa. Joseneuza Aguiar

Materiais do prof. Salvador Alves

<https://blog.geekhunter.com.br/voce-conhece-html-semantic/>

<https://diegomariano.com/componentes-bootstrap/>

<https://www.w3schools.com/>

<https://developer.mozilla.org/pt-BR/>

Sites Úteis:

<https://htmlcheatsheet.com/> ; <https://websitesetup.org/css3-cheat-sheet/> -

CheatSheet's de HTML e CSS. Uma cheat sheet é um documento rápido para lembrar informações.

<https://www.lipsum.com/feed/html> - Profissionalmente, se usa Lorem Ipsum para se preencher textos de teste.

<https://getbootstrap.com/> - Site do bootstrap (Framework CSS)

<https://rockcontent.com/br/talent-blog/frameworks-css/> - Bootstrap possui vários concorrentes.

<https://codepen.io/trending> - Desenvolvimento de sites online.

<https://pixabay.com/pt/> e <https://unsplash.com/> - Imagens gratuitas e livres de direito autoral.

<https://www.free-css.com/> - Templates de sites livre de direitos autorais.

<https://css-tricks.com/> - Site que mostram códigos CSS para estudo.

<https://br.000webhost.com/> - Hospedagem de site grátis

<https://www.w3schools.com/html/default.asp> ; <https://developer.mozilla.org/pt-BR/> ;

- Sites que ensinam muito sobre desenvolvimento web (HTML, CSS, JS)