

HTML

Tags:

- Abertura-> Ex: <p>
- Fechamento-> Ex: </p>
- Conteúdo-> O que fica entre a abertura e o fechamento
- Atributos-> Dá características para as tags (class, id)
- Sempre as tags com letras minúsculas

Elementos com Conteúdo

Ex: <h1>, <p>

Elementos sem Conteúdo

Ex: <input>,

src -> caminho até o conteúdo

href-> link de acesso

Estrutura de uma página HTML:

Todos o código precisa estar dentro da estrutura HTML:

<html>

 <head>

 Configurações não visíveis da página e título da página, links para CSS, Js entre outros para conectar ao html

 </head>

 <body>

 Onde fica armazenado o conteúdo visível da página

 </body>

</html>

SEO:

Técnicas para ranquear melhor meu site nos sites de buscas:
Importante usar corretamente as tags para organizar o site.

Principais Tags para Textos:

<h1> Títulos: Por ordem de importância, <h1> para a maior importância até <h6> de menor importância.

<h1></h1>

<h2></h2>

<h3></h3>

<h4></h4>

<h5></h5>

<h6></h6>

*recomendado ter apenas um tag <h1> por página.

<p> Parágrafos: Para criar um parágrafo dentro do site com a tag <p>.

*Posso usar quantas tags <p> for necessário

**
 Pular linhas:** Para pular uma linha se usa a tag sem conteúdo

** Deixar em negrito dentro do parágrafo:** se usa a tag em volta da palavra/ frase que quero estilizar.

<i></i> Deixar em itálico dentro do parágrafo: se usa a tag <i></i> em volta da palavra/ frase que quero estilizar.

** Deixar a palavra/frase forte:** Para dar ênfase e deixar em negrito para os mecanismos de buscas no meu site.

Navegando entre Pastas:

Posso armazenar dentro do scr tanto uma imagem disponível na internet quanto percorrer dentro das pastas do projeto para encontrar o conteúdo que quero trazer para minha página, recomendado sempre salvar o conteúdo dentro da mesma pasta do projeto do site, mas criar subpastas para organização (HTML, CSS, JS, ...).

Usar ../ para percorrer as pastas para encontrar o conteúdo.

Links:

`<a>` Texto para armazenar o link ``

`href` = "armazena o caminho/url para o link a ser acessado "

`target` = para onde enviar a pessoa, usar o `_blank` para abrir uma nova aba até o link

`mailto` = usar no href para acessar diretamente para enviar um e-mail.

`tel`= usar no href para acessar uma ligação para um telefone, usar código do país e ddd (+5515996820177)

`id`= usar no href para acessar determinado local dentro da própria página, Ex: um parágrafo, adicionamos um id no parágrafo e depois colocamos `#id` dentro do href

Ex:

`target`:

`` LinkedIn ``

`mailto`:

`` Envie um e-mail ``

`tel`:

`` Me ligue! ``

`id`:

`` Vá para o fim ``
`<p id="fim">` Esse é o fim `</p>`

***Também podemos chamar outro arquivo dentro do mesmo projeto através do href= "../. index.html (caminho para o acesso)" .**

Listas:

`` Lista Ordenada: Inicia uma lista ordenada, e dentro dele tem suas ``Item``, para registrar seus itens.

Use a lista ordenada quando a sequência dos itens é **FUNDAMENTAL** para manter a sequência lógica. Mudar a ordem deles estraga o sentido, pois a numeração (1, 2, 3...) faz parte da informação.

Ex:

<h1>4 série B</h1>

```
<ol>
  <li>Item1</li>
  <li>Item2</li>
  <li>Item3</li>
</ol>
```

). Você pode usar alguns atributos para mudar o comportamento dela:

- **start:** Faz a lista começar de um número diferente de 1.
 - <ol start="5"> (A lista começaria em: 5, 6, 7...)
- **reversed:** Faz a contagem ser regressiva.
 - <ol reversed> (Uma lista de 3 itens apareceria como: 3, 2, 1)
- **type:** Muda o tipo de marcador.
 - type="A" (Para letras maiúsculas: A, B, C...)
 - type="a" (Para letras minúsculas: a, b, c...)
 - type="I" (Para números romanos maiúsculos: I, II, III...)
 - type="i" (Para números romanos minúsculos: i, ii, iii...)

** Lista Não Ordenada:** Inicia uma lista não ordenada, e dentro dele tem suas Item, para registrar seus itens.

Use a lista **não ordenada** quando a sequência dos itens **NÃO é importante**. Se você mudar a ordem deles, a lista continua fazendo sentido e não perde a informação principal.

Ex:

<h1>Lista de Compras</h1>

```
<ul>
  <li>Arroz</li>
  <li>Feijão</li>
  <li>Cenoura</li>
</ul>
```

Elementos Estruturais:

<div> </div> = Ela é uma caixa que armazena meus conteúdos(tag's). Ela é invisível, mas serve para a separação e organização de conteúdo, a organização das divs serão mais usadas durante o CSS, para estilizar todo o site a partir delas.

** ** = É uma pequena "div" dentro do meu texto, quando for estilizar pegamos o id do span e podemos estilizar apenas o que está dentro do span, seja cor, tamanho, fonte, etc.

Os Principais Elementos Estruturais Semânticos

Aqui estão os substitutos modernos para as antigas <div id="header">, <div id="menu">, etc.

<header>: O cabeçalho. Geralmente contém o logo, o título do site e a navegação principal.

<nav>: A área de navegação. Usada para agrupar os links principais do site (o menu).

Geralmente, o menu principal com os links mais importantes fica logo no topo. A tag <nav> serve para agrupar **apenas** esses links de navegação principal.

<main>: O conteúdo principal e único daquela página. Deve existir **apenas um** <main> por página.

Esta é a tag mais importante para o conteúdo **único** da página atual. Ela diz: "Ignore o cabeçalho e o rodapé que se repetem, o motivo pelo qual o usuário está *nesta* página está aqui dentro".

<section>: Uma seção genérica de conteúdo. Pense nela como um capítulo de um livro. Geralmente tem um título (<h2>, <h3>, etc.).

Um "Capítulo" do Conteúdo. É um agrupamento de conteúdo sobre o mesmo tema. Diferente do <article>, uma <section> pode não fazer sentido se for "recortada" e lida isoladamente.

<article>: Um bloco de conteúdo completo e independente que faria sentido por si só, fora da página (Ex: um post de blog, uma notícia, um produto em uma lista de produtos).

<aside>: Para conteúdo secundário, relacionado ao conteúdo principal, mas não essencial. Uma barra lateral (sidebar) é o exemplo mais comum.

Conteúdo prático: Uma pequena biografia do autor, links para "posts relacionados", uma lista de categorias, publicidade. Geralmente, é o que fica na barra lateral (sidebar).

Por que não uma <div>? Você está dizendo ao navegador: "Esta informação é extra, um complemento. O mais importante está no <article>".

<footer>: O rodapé. Geralmente contém informações de copyright, links para redes sociais e informações

Imagens no HTML:

****: A tag **img** armazena uma imagem, seja ela online, dentro do seu computador ou dentro do projeto. Podemos também definir a altura da imagem com o **height** da imagem e o **width**. Mas tomar cuidado ao mexer nos tamanhos para não distorcer a imagem, se usa ou **height** ou **width**, somente um deles, mas atualmente se usa o CSS para quando necessário definir o tamanho da imagem.

src = "/caminho-da-imagem" : (Source) dentro do **src** fica o caminho para imagem.

alt="descrição-imagem" : (Alternative Text) Descreve o conteúdo da imagem.

Ele não é opcional! Acessibilidade: Leitores de tela para pessoas com deficiência visual leem o texto do atributo **alt** em voz alta. Sem ele, a pessoa não tem como saber o que a imagem representa. SEO (Google): O Google não "vê" imagens, ele "lê" o texto **alt** para entender sobre o que é a imagem e o contexto da sua página, ajudando no ranking das buscas. Links Quebrados: Se a imagem não carregar por algum motivo, o texto do **alt** será exibido no lugar.

<figure> </figure>: Usamos a tag **figure**, com a tag **img** dentro nos dá algumas ferramentas como uma legenda para imagem com **<figcaption></figcaption>**.

Ex:

```
<figure>
  

  <figcaption>Legenda da Imagem</figcaption>
</figure>
```

Tabelas:

<table> </table>: Ela é a responsável por criar a tabela dentro do HTML.

<tr></tr>: (**Table Row**) Esta é a tag que **cria as LINHAS**. Cada vez que você usa **<tr>**

<th> </th> : (**Table Header**) Esta é a tag que cria as **CÉLULAS de cabeçalho**, que são como **<td>**s, mas para os títulos das colunas.

<td> </td> : (**Table Data**): Esta é a tag que cria as **CÉLULAS de dados** dentro de uma linha.

Ex:

```
<table>
  <tr>
    <th>cabeçalho1</th>
    <th>cabeçalho2</th>
    <th>cabeçalho3</th>
  </tr>

  <tr>
    <td>linha1</td>
    <td>linha2</td>
    <td>linha3</td>
  </tr>

  <tr>
    <td>linha1</td>
    <td>linha2</td>
    <td>linha3</td>
  </tr>

  <tr>
    <td>linha1</td>
    <td>linha2</td>
    <td>linha3</td>
  </tr>
```

***a cada <tr><td></td></tr> se cria uma nova linha na tabela**

Ex. visual da tabela seguindo o código acima:

<th>	<th>	<th>
<td>	<td>	<td>
<td>	<td>	<td>
<td>	<td>	<td>

***th cabeçalho da coluna**

***td linha da coluna**

A Estrutura Semântica: <thead>, <tbody> e <tfoot>

Assim como a página tem <header> e <footer>, as tabelas também têm tags para separar o cabeçalho, o corpo e o rodapé. Usá-las é uma excelente prática.

- **<thead>**: Use para envolver a(s) linha(s) do **cabeçalho** (aquelas que contêm as tags <th>).
- **<tbody>**: Use para envolver todas as linhas do **corpo** principal da tabela (aquelas com os dados em <td>).
- **<tfoot>**: Use para envolver a(s) linha(s) do **rodapé** da tabela (pode conter resumos, totais, etc.). Não é sempre usado, mas é bom saber que existe.

Por que usar isso?

- **Organização**: Seu código fica muito mais claro.
- **Estilização (CSS)**: Fica mais fácil estilizar o cabeçalho de forma diferente do corpo.
- **Funcionalidade**: Em tabelas muito longas que são impressas em várias páginas, o navegador pode repetir o <thead> em cada nova página, facilitando a leitura.

Tabelas no HTML

- **<table>**: A tag principal que envolve toda a estrutura da tabela.
- **<thead>**: Agrupa o conteúdo do cabeçalho da tabela.
- **<tbody>**: Agrupa o conteúdo do corpo principal da tabela.
- **<tfoot>**: Agrupa o conteúdo do rodapé da tabela.
- **<tr> (Table Row)**: Cria uma **linha** dentro da tabela.
- **<th> (Table Header)**: Cria uma **célula de cabeçalho** (o título de uma coluna). Fica dentro de uma <tr>.
- **<td> (Table Data)**: Cria uma **célula de dados** padrão. Fica dentro de uma <tr>.

Ex:

```
<table>

<thead>
  <tr>
    <th>Nome do Produto</th>
    <th>Quantidade</th>
    <th>Preço Unitário</th>
  </tr>
</thead>

<tbody>
  <tr>
    <td>Caneta Azul</td>
    <td>10</td>
    <td>R$ 1,50</td>
  </tr>

  <tr>
    <td>Caderno 10 Matérias</td>
    <td>2</td>
    <td>R$ 25,00</td>
  </tr>

  <tr>
    <td>Borracha</td>
    <td>5</td>
    <td>R$ 2,00</td>
  </tr>
</tbody>

<tfoot>
  <tr>
    <td>Total</td>
    <td>17 itens</td>
    <td>--</td>
  </tr>
</tfoot>

</table>
```

Videos em HTML:

<video

src="/caminho-do-video"

controls

controlsList = "nodownload"

autoplay

muted

loop

poster = "/caminho-para-capa"

></video > : A tag video armazena um video, seja ela online, dentro do seu computador ou dentro do projeto.

src = "/caminho-do-video" : (Source) dentro do src fica o caminho para o video.

controls : Para o video possa ter os botões de comando(play, stop, volume, etc)

controlsList = "nodownload" : Para remover a opção de baixar o vídeo.

autoplay : Inicia automaticamente o vídeo, mas algumas navegadores bloqueiam por conta do som, nesse caso usamos o muted, pois remove o som e podemos ativar o autoplay assim que acessar a página.

poster : Para colocar uma capa antes de iniciar o vídeo.

loop: Faz com que o vídeo recomece automaticamente assim que termina. É ótimo para vídeos curtos de fundo ou decorativos.

Também podemos usar de outra forma: para dar mais opções caso ocorra algum problema com o vídeo.

<video>

<source src = "/caminho-do-video.mkv"> //primeira opção de vídeo

<source src = "/caminho-do-video.mp4"> //segunda opção de vídeo

<source src = "/caminho-do-video.mov"> //terceira opção de vídeo

</video>

Iframe:

<iframe src = "caminho-para-site"> </iframe> : Adiciona um site dentro do nosso site. Cria uma caixa com acesso direto para outro site, ex.: Quando tem um site de uma loja que coloca a localização do google dentro dele, ou um video do you tube. Não são todos os sites que estão disponíveis. Podemos definir o tamanho da caixa com width(largura) e height(altura).

Input pt1:

<label>

Rua:

<input

autocomplete = "completa automaticamente"

disabled

readonly

value = "valor"

placeholder = "Digite seu nome"

autofocus

> : Elas são aquelas pequenas caixas em um site para receber informações. Ex: a barra de pesquisa do Google.

</label>

autocomplete = "define-informação": defino qual informação vai receber para completar, seja ela, nome, endereço, telefone, cidade, etc.

disabled : Quando quero manter ele desabilitado por algum motivo.

readonly : Quando deixo habilitado mas impossibilitado de digitar no input.

value = "valor": Quando quero determinar um valor para dentro do input.

placeholder = "Digite seu nome": Define um texto de fundo quando o input está vazio.

autofocus : Começa a pagina já focado no input apontado. * Cuidado para não colocar um autofocus em mais de um input, recomendado é usar um autofocus por pagina.

<label> </label> : Define o nome para o input.

Outra forma de usar o label: Usamos o for dentro da label e o id dentro do input para criar a ligação entre eles.

```
<label for = "rua">  
  Rua:  
  <input id = "rua"  
    autocomplete = "completa automaticamente"  
    disabled  
    readonly  
    value = "valor"  
    placeholder = "Digite seu nome"  
    autofocus >  
</label>
```

Input pt2:

<input type = " " > :

type = "color" : Ele abre uma caixa para o usuário escolher uma cor em RGB

type = "range" min = "0" max = "100" value = "50" : Define o estilo/formato do input, podemos definir o range mínimo e máximo e o valor que se inicia

type = "password" : Define que vai receber uma senha, então substitui os valores digitados visíveis por bolinhas para aumentar a segurança

type = "email" : Define que vai receber um e-mail, por obrigação precisa ter um '@' e um '.'

type = "checkbox" : Define que vai ser uma caixinha daqueles de ticar para confirmar algo

type = "file" multiple accept = ".jpeg" : Para fazer o upload de um arquivo, **multiple** para selecionar vários arquivos ao mesmo tempo, **accept** define o formato do arquivo, o accept pode definir o tipo (video/*, image/*, audio/*)

type = "number" : Define que irá receber números

type = "date" : Define que irá receber uma data

type = "datetime-local" : Define que irá receber data + horário

type = "tel" : Define que vai receber um telefone

type = "url" : Define que vai receber uma URL

<label for = facebook> Facebook </label>

<input

type = "radio" id = "facebook" name = "redes-sociais" :

Define aquela bolinha de escolher uma opção, e p id define qual será a opção, e o name para relacionar elas

>

<label for = instagran> Instagra </label>

<input

type = "radio" id = "instagran" name = "redes-sociais" :

Define aquela bolinha de escolher uma opção, e p id define qual será a opção, e o name para relacionar elas

>

<label for = twitter> Twitter </label>

<input

type = "radio" id = "twitter" name = "redes-sociais" : Define aquela bolinha de escolher uma opção, e p id define qual será a opção, e o name para relacionar elas

>

***se todos os radios tiverem o mesmo name, só pode se escolher uma opção, para estarem interligados precisam do mesmo name**

Forms:

<form> </form> : É o contêiner que agrupa campos de entrada para coletar e enviar dados a um servidor.

Capturar dados de entrada

Input -. Entrada

Action -> Define para qual url será enviado os dados, vazio se para a mesma url

Method

- **GET**
- **POST**

```

<form action = " " method = " " ***enctype = "multipart/form-data">
  <label> Rua:
    <input type = "texto" name = "rua" autocomplete = "street-
      address">
  </label>

  <label> Número:
    <input type = "number" name = "numero">
  </label>

  ***<label> Foto:
    <input type = "file" name = "arquivo">
  </label>

  <button type = "reset"> Limpar </button>
  <button type = "submit"> Enviar Dados </button>

</form>

```

Ex:

A estrutura começa com a tag **<form>** envolvendo todos os campos de coletas, podemos acrescentar o **<label>** para definirmos o que estamos esperando receber no **<input>**, definimos o **<type>** do input para definir o tipo de dados recebidos, podemos usar caso necessário o **<autocomplete>**, fechamos a **</label>** e precisamos criar um **<button>** do tipo **"submit"** para podermos enviar as informações coletadas, também podemos ter um **<button>** do tipo **"reset"** para apagar todos os campos digitados, e depois fechamos com a tag **</form>**, podemos criar quantos inputs precisarmos dentro de um forms.

action = " " : Define a URL de destino para onde os dados do formulário serão enviados. Se deixado vazio, envia para a própria página atual.

method = " " : Define o método HTTP para o envio. (GET/POST)

***GET** : Os dados são anexados à URL,

- **Quando usar?** Para ações que **não modificam dados** no servidor. Ideal para buscas, filtros, ou quando você quer que a URL possa ser salva nos favoritos ou compartilhada com os resultados já visíveis.
- **Limitações:** Tem um limite de tamanho e **não deve JAMAIS** ser usado para enviar dados sensíveis (como senhas), pois a informação fica visível na URL e no histórico do navegador.

***POST** : Os dados são enviados no "corpo" da requisição HTTP, de forma invisível para o usuário na barra de endereço.

- **Quando usar?** Para ações que **modificam dados** no servidor. Ideal para cadastros, logins, envio de comentários, uploads de arquivos, ou qualquer envio de dados sensíveis.

- **Limitações:** Não tem limite prático de tamanho de dados. A página resultante de um POST não deve ser salva nos favoritos (se você atualizar a página, o navegador geralmente pergunta se você quer "reenviar os dados do formulário").

enctype = "multipart/form-data" : Informa ao servidor que irá enviar um arquivo

*****Opcional, só se usa quando precisar enviar um arquivo de qualquer tipo pelo forms**

Button:

`<button type = "reset"> Limpar </button>`

`<button type = "submit"> Enviar Dados </button>`

`<button type = "button" onclick = "chama o js para realizar uma ação ao clicar"> Menu </button>`

type = "reset" : Reseta o que for digitado

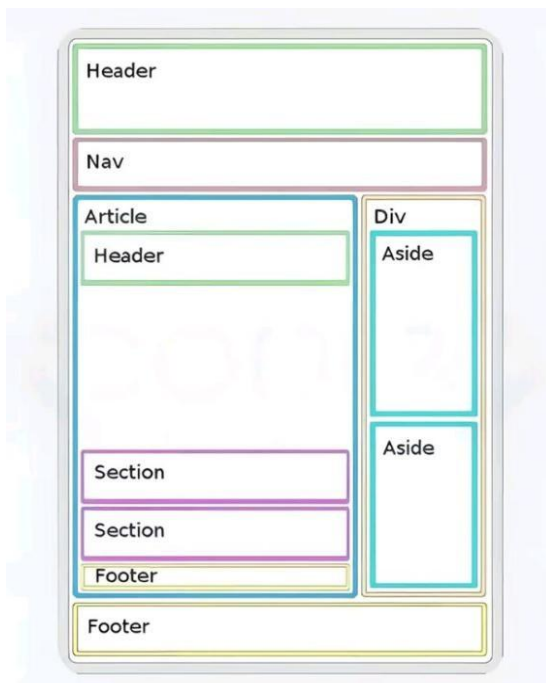
type = "submit" : Para enviar as informações coletadas

type = "button": Cria um botão normal para realizar alguma ação

disabled = desativa o botão

onclick = define a ação a ser realizada dentro do js

HTML Semântico:



- **<header>**: O cabeçalho, seja do site todo (topo) ou de uma parte específica (como o header dentro do article).
- **<nav>**: A área de navegação principal.
- **<article>**: O conteúdo principal, que faz sentido por si só (um post de blog, uma notícia).
- **<section>**: As diferentes "seções" ou "capítulos" dentro do seu artigo.
- **<aside>**: Uma barra lateral com conteúdo relacionado, mas não essencial (posts relacionados, publicidade).
- **<footer>**: O rodapé, seja do site todo (embaixo) ou do artigo (com informações do autor, por exemplo).
- **<div>**: Uma caixa genérica, sem significado, usada apenas para agrupar elementos para facilitar a estilização com CSS (como criar a coluna da direita no diagrama).

Ex Código:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Minha Página Semântica</title>
</head>
<body>

  <header>
    <h1>Logo do Meu Site</h1>
  </header>

  <nav>
    <a href="#">Home</a>
    <a href="#">Sobre</a>
    <a href="#">Contato</a>
  </nav>

  <main>

    <article>
      <header>
        <h2>Título do Meu Artigo Fantástico</h2>
      </header>

      <section>
        <h3>Introdução</h3>
        <p>Este é o primeiro parágrafo do meu artigo...</p>
      </section>

      <section>
        <h3>Desenvolvimento</h3>
        <p>Aqui eu desenvolvo mais o assunto...</p>
      </section>

      <footer>
        <p>Postado por: Seu Nome</p>
      </footer>
    </article>

    <aside>
      <h3>Posts Relacionados</h3>
      <ul>
        <li><a href="#">Outro Post</a></li>
        <li><a href="#">Mais um Post</a></li>
      </ul>
    </aside>

  </main>

  <footer>
    <p>&copy; 2025 - Todos os direitos reservados.</p>
  </footer>

</body>
</html>
```