

Índice

Introdução a HTML	1
DocType HTML:	2
HTML Lang - Idioma	2
<head> Cabeçalho	3
Tags do tipo <meta>	3
<Title>	5
<body>	5
<h1>, <h2>, <h3>, <h4>, <h5> e <h6>	5
<p>	6
<div>	6
<script>	6
window.onload:	7
JS arquivo externo	7
Adicionando CSS	9
<style> Esta é uma tag HTML usada para definir estilos de formatação, como cores, tamanhos de fonte, espaçamento e outros aspectos visuais de elementos HTML.	9
Listas	10
Uma boa pesquisa no google deve conter :	11
Links e afins	12
Dúvidas	12

Assuntos: Introdução a HTML

Abertura Alunos:

Apresentação pitch
Yuri e Daniel.

Front-end

No front-end geralmente se usam 3 ferramentas: HTML, CSS e JavaScript.

Sendo elas, respectivamente, duas linguagens de marcação, ou seja, linguagens que marcam o que se tem na parte da frente do site, como onde e como ficam textos, imagens, links, e etc. E a última, uma linguagem de programação, capaz de executar comandos em um site.

HTML é uma linguagem de marcação, utilizada na construção de páginas web. Documentos HTML podem ser interpretados por navegadores.

Cada página HTML consiste em uma série de tags (ou elementos) que podem ser consideradas blocos de construção de páginas.

A linguagem HTML é sempre lida de forma sequencial.

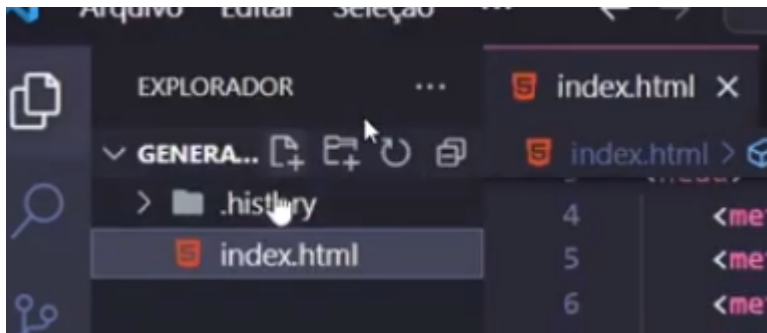
Introdução a HTML

A IDE utilizada é o VsCode.

No VsCode, é possível iniciar a escrita de uma tag, e ela ser autocompletada.

O próprio VsCode quando inicializado possui uma estrutura padrão para HTML5, que assim que identifica que estamos utilizando a extensão **.html** ou com o **!**.

A primeira página sempre terá o nome de Index, a estrutura base é uma versão da linguagem de marcação que a gente vai usar com as informações e estruturas mais atuais e com facilidade em comandos semânticos.



`<!-- -->` é assim que é feito comentários dentro do VsCode.

Comandos em html geralmente se escrevem entre `<>`.

Para indicar o início de um elemento, e depois, seu final, utilizamos respectivamente

`<elemento>` conteúdo do elemento `</elemento>`.

Perceba como para fechar um comando, se coloca **`</elemento>`**, com o sinal de **/** na frente, semelhante a abrir e fechar colchetes **{ }** em java.

DocType HTML:

O DOCTYPE html é uma declaração que informa ao navegador qual versão do HTML está sendo usada em um documento. No HTML5, a declaração DOCTYPE é simplesmente:

```
1 <!DOCTYPE html>
```

Esta declaração informa ao navegador que o documento está em HTML5. O navegador então processa e exibe a página de acordo com as regras do HTML5.

No HTML 4.01, a declaração DOCTYPE poderia ser uma das seguintes:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Cada uma dessas declarações DOCTYPE tinha um caso de uso diferente, como páginas web estritas, páginas web com elementos de transição e páginas web com quadros.

No entanto, com o HTML5, a declaração DOCTYPE tornou-se muito mais simples e não requer uma referência a uma DTD. Isso ocorre porque o HTML5 não se baseia mais na linguagem de marcação generalizada (SGML) e, portanto, não precisa de uma referência a uma DTD.

Em resumo, o DOCTYPE html é uma declaração que informa ao navegador qual versão do HTML está sendo usada em um documento.

HTML Lang - Idioma

O atributo lang em HTML serve para definir o idioma de um elemento, podendo ser usado tanto em todo o documento como em partes dele. A declaração do idioma do conteúdo é importante para a acessibilidade da página, pois ajuda leitores de tela e outros meios de acesso a identificar o idioma do texto e aplicar as configurações corretas de pronúncia e acentuação. Além disso, também é útil para motores de busca e ferramentas de tradução automática.

Exp:

```
1 <!DOCTYPE html>  
2 <html lang="pt-BR">
```

Neste exemplo, o atributo lang é usado para definir o idioma do documento como português do Brasil (pt-BR). Em seguida, é usado em um elemento <blockquote> para definir que o conteúdo desse elemento está em inglês (en). Isso é útil para leitores de tela e outros meios

de acesso identificarem o idioma do conteúdo e aplicarem as configurações corretas de pronúncia e acentuação.

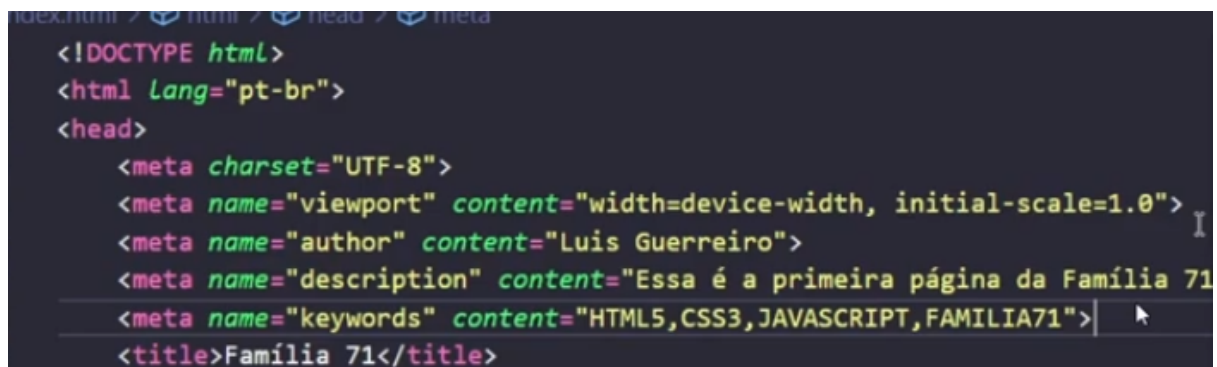
<head> Cabeçalho

O elemento <head> em HTML é usado para conter informações sobre o documento, como o título, metadados, links para arquivos CSS e scripts JavaScript. Essas informações são usadas pelo navegador para exibir a página corretamente e não são exibidas no corpo do documento.

A importância do elemento <head> em HTML para pesquisas no Google é que ele contém informações importantes sobre o documento que podem ajudar o motor de busca a entender e indexar o conteúdo da página.

<meta>

As tags do tipo <meta> em HTML são usadas para fornecer metadados (Metadados são dados que descrevem outros dados, que não necessariamente serão apresentados para o meu usuário) sobre o documento, como informações sobre o autor, charset, viewport, e outras informações que não são exibidas no corpo do documento. Essas tags são inseridas no elemento <head> do documento.

A screenshot of a code editor showing HTML code. The code is as follows:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="author" content="Luis Guerreiro">
  <meta name="description" content="Essa é a primeira página da Família 71">
  <meta name="keywords" content="HTML5,CSS3,JAVASCRIPT,FAMILIA71">
  <title>Família 71</title>
```

Neste exemplo, as tags <meta> são usadas para fornecer informações sobre o charset, viewport, autor, descrição e palavras-chave do documento. O atributo name é usado para definir o tipo de metadado e o atributo content é usado para fornecer o valor do metadado.

Algumas outras propriedades comuns das tags <meta> incluem:

- **http-equiv:** Define um cabeçalho HTTP para o servidor web.
- **content:** Define o valor do cabeçalho HTTP ou metadado.
- **scheme:** Define o esquema de codificação do valor do metadado.
- **viewport:** é usado para controlar a visualização visual da página da web em dispositivos móveis. A meta tag viewport permite que os desenvolvedores definam a

largura e a escala da página da web para otimizá-la para diferentes tamanhos e resoluções de tela.

- **keywords:** são uma meta tag que fornece palavras-chave ou frases-chave que descrevem o conteúdo da página, que são usadas por mecanismos de busca.
- **charset="UTF-8"** : é uma declaração para garantir que o navegador interprete corretamente a codificação de caracteres do documento HTML como UTF-8, o que é crucial para exibir corretamente texto em vários idiomas e caracteres especiais.

PS: É importante notar que os principais mecanismos de busca, como o Google, não consideram mais a meta tag de keywords como um fator significativo para classificar as páginas nos resultados de pesquisa. Eles agora confiam principalmente no conteúdo real da página e em outros sinais para determinar a relevância.

As tags <meta> são importantes para a indexação do site em motores de busca, acessibilidade e para definir as configurações de visualização do documento em dispositivos móveis.

Informações adicionais:

O meta charset existe para que quando um navegador encontra essa tag no cabeçalho (<head>) de um documento HTML, ele sabe interpretar o conteúdo do documento como sendo codificado em UTF-8. Isso é essencial para garantir que caracteres especiais, acentos, símbolos e caracteres de idiomas não-ocidentais sejam exibidos corretamente na página da web.

Por exemplo, se você tiver texto em japonês, russo, árabe ou qualquer outro idioma que não seja inglês, e não especificar a codificação correta, esses caracteres podem ser exibidos incorretamente no navegador.

<Title>

A tag <title> em HTML é usada para especificar o título de uma página da web. O título é exibido na aba ou janela do navegador e também é usado pelos mecanismos de busca para ajudar os usuários a entender do que se trata a página. O título deve refletir com precisão o conteúdo da página e ser exclusivo para cada página do site

```
<title>Família 71</title>
```

<body>

É uma tag que define o conteúdo principal de uma página da web. Todo o conteúdo visível para os usuários, como texto, imagens, vídeos, links e outros elementos, é colocado dentro da tag <body>.

<h1>, <h2>, <h3>, <h4>, <h5> e <h6>

As tags <h1>, <h2>, <h3>, <h4>, <h5> e <h6> em HTML são usadas para definir diferentes níveis de cabeçalhos ou títulos em uma página da web. Cada uma delas representa um nível de importância diferente, com <h1> sendo o mais importante e <h6> o menos importante. Essas tags são usadas para estruturar o conteúdo da página e fornecer uma hierarquia clara de títulos, mas sua principal característica é a diferença de tamanho H1 sendo a maior e H6 sendo a menor:

- **<h1>**: Define o cabeçalho de nível 1, o título principal da página.
- **<h2>**: Define o cabeçalho de nível 2, usado para títulos secundários ou subseções importantes.
- **<h3>**: Define o cabeçalho de nível 3, usado para títulos de terceiro nível, normalmente subordinados aos <h2>.
- **<h4>**: Define o cabeçalho de nível 4, usado para títulos de quarto nível, e assim por diante.
- **<h5>**: Define o cabeçalho de nível 5.
- **<h6>**: Define o cabeçalho de nível 6, sendo o menos importante dos cabeçalhos.

A hierarquia de cabeçalhos é importante não apenas para a estrutura visual da página, mas também para a acessibilidade e SEO (Search Engine Optimization). Os mecanismos de busca consideram a hierarquia de cabeçalhos ao indexar o conteúdo da página, e os leitores de tela utilizados por pessoas com deficiência visual usam essa hierarquia para navegar pelo conteúdo da página de forma mais eficiente. Pode-se ter mais de um <h2>, ou <h3>, etc, mas jamais mais de um <h1> em uma página.

<p>

A tag <p> em HTML é usada para definir um parágrafo de texto. Ela é usada para segmentar e estruturar o texto em blocos coesos em uma página da web. Qualquer texto colocado entre a abertura <p> e o fechamento </p> é considerado um parágrafo separado.

Lorem Ipsum

É um texto fictício amplamente utilizado como espaço reservado em projetos de design gráfico, tipografia, web design e desenvolvimento de sites. Ele é empregado como um "texto padrão" quando o conteúdo real não está disponível ou não é relevante para o propósito imediato.

Seu uso é principalmente para preencher espaços em um layout de página para que os designers possam avaliar a aparência visual do design sem se distrair com o conteúdo real.

O Lorem Ipsum tem uma distribuição de letras e palavras que se assemelha ao texto real, o que ajuda a dar uma ideia mais precisa do layout final da página. Geralmente, é utilizado parágrafos longos, mas também existem variações com frases mais curtas ou até mesmo com palavras individuais repetidas.

Este texto é amplamente utilizado como um "espaço reservado" em projetos de design e desenvolvimento de sites.

Exemplo de Lorem Ipsum:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

<div>

A tag <div> em HTML é um elemento de bloco usado para criar divisões ou seções dentro de uma página da web. Ela é frequentemente usada para agrupar e organizar outros elementos HTML, como texto, imagens, formulários, e outros elementos HTML, permitindo que você os estilize ou manipule como um grupo usando CSS (Cascading Style Sheets) ou JavaScript.

<script>

A tag <script> em HTML é usada para incorporar scripts ou programas de script em uma página da web. O atributo type da tag <script> especifica o tipo de conteúdo do script.

Anteriormente, o atributo type era usado para especificar o tipo de linguagem de script sendo usado, como JavaScript (type="text/javascript"), VBScript (type="text/vbscript") ou outras linguagens. No entanto, em HTML5, o atributo type é considerado opcional para scripts em JavaScript, pois o tipo padrão é text/javascript.

window.onload:

window.onload é um evento em JavaScript que ocorre quando a página HTML completa seu carregamento, incluindo todo o conteúdo (imagens, scripts, folhas de estilo, etc.). Ele é disparado quando o objeto window, que representa a janela do navegador, termina de carregar todos os recursos da página.

Quando window.onload é acionado, isso indica que é seguro começar a interagir com os elementos da página ou executar operações que dependem do carregamento completo da página.

O function do código está sendo usada para criar uma função anônima.

{alert ("Estamos criando um alerta");} é o que será exibido para o usuário.

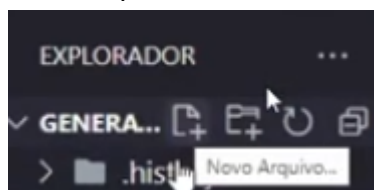
Aqui é uma inserção direta no código:

```
<title>Família 71</title>
<script type="text/JAVASCRIPT">
  window.onload = function() {
    alert("Estamos criando uma janela de alerta!!!");
  }
</script>
```

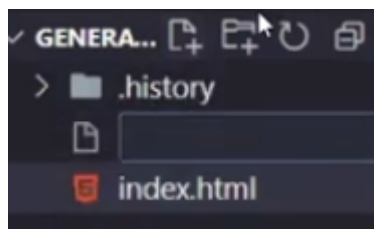
JS arquivo externo

O mais indicado para trabalhar é ter um arquivo externo para chamar uma execução JS dentro do seu código HTML.

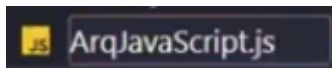
Vá no explorador do VsCode em Novo Arquivo:



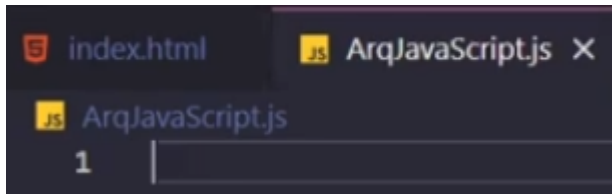
Cria um documento que você tem nomear:



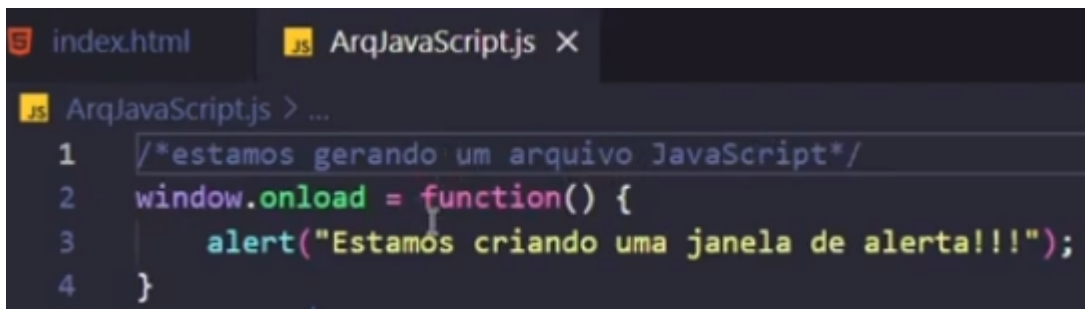
Ao nomeá-lo é necessário adicionar a extensão **.js**.



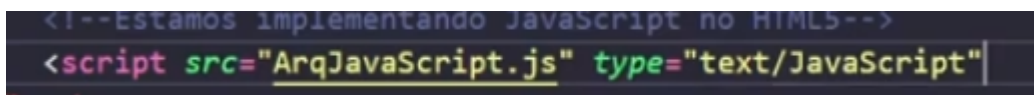
Assim que criado é necessário montá-lo pois ele abre por padrão sem nenhuma informação.



E assim você começa a codar em JS para ser executado no Index:



Para executar no Index, dentro do <head> deve ser criado outro **<script src = “ XXXX nome do arquivo”** indicando o caminho e deve ser declarado qual tipo de linguagem.



src="nome do arquivo": O atributo src especifica o caminho do arquivo JavaScript que será incluído na página. Este caminho pode ser uma URL (se o arquivo estiver hospedado em um servidor remoto) ou um caminho relativo ou absoluto para um arquivo no sistema de arquivos local. Por exemplo, src="meu_script.js".

type="text/javascript": O atributo type especifica o tipo de conteúdo do script. No passado, quando diferentes tipos de scripts eram mais comuns na web, esse atributo era usado para indicar a linguagem de script sendo usada. No caso de JavaScript, o valor text/javascript é especificado. Embora em HTML5, este atributo seja opcional para scripts JavaScript, pois text/javascript é o tipo padrão.

Então, quando você coloca <script src="nome do arquivo" type="text/javascript"></script> em seu código HTML:

- O navegador encontra essa tag <script> e começa a carregar o arquivo JavaScript especificado no atributo src.
- Quando o arquivo é totalmente carregado, o navegador executa o código JavaScript contido nele.

- Se houver algum erro no código JavaScript ou o arquivo não for encontrado, o navegador pode exibir mensagens de erro no console do desenvolvedor.

Adicionando CSS

O CSS é uma linguagem de estilo poderosa que permite controlar a aparência e o layout de uma página da web de forma flexível e eficiente.

<style> Esta é uma tag HTML usada para definir estilos de formatação, como cores, tamanhos de fonte, espaçamento e outros aspectos visuais de elementos HTML.

type="text/css": O atributo type especifica o tipo de conteúdo dentro da tag <style>. Neste caso, text/css indica que o conteúdo dentro da tag <style> é um código CSS (Cascading Style Sheets).

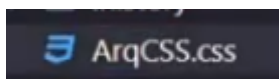
body: está declarando que tudo dentro da página será definido por uma cor.

Implementação no código:

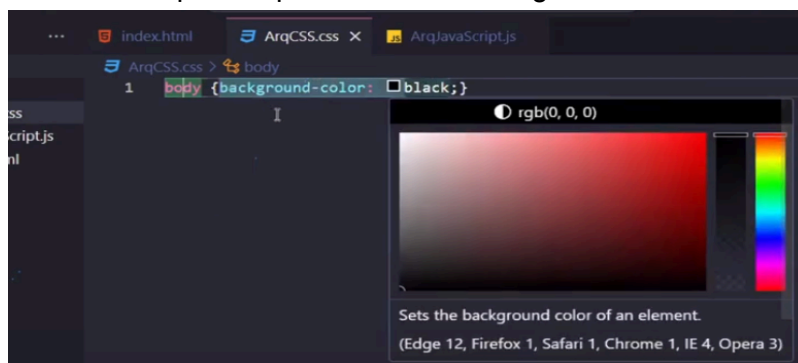
```
<style type="text/Css">
  body {color: blue}
```

Arquivo externo:

Segue os passos da implementação de JS, só que utilizando a extensão CSS.



Dentro do arquivo implementando o código:



Dentro do index nós temos o código abaixo:

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

<link>: Esta é uma tag HTML usada para estabelecer uma conexão entre o documento atual (página HTML) e um recurso externo, como uma folha de estilo CSS.

rel="stylesheet": O atributo *rel* especifica o relacionamento entre o documento atual e o recurso externo que está sendo vinculado. Neste caso, *stylesheet* indica que o recurso externo é uma folha de estilo que deve ser aplicada ao documento.

type="text/css": O atributo *type* especifica o tipo de conteúdo do recurso externo. Neste caso, *text/css* indica que o conteúdo é uma folha de estilo CSS.

href="nome do arquivo em CSS": O atributo *href* especifica o local do arquivo CSS externo que será vinculado ao documento. Este atributo contém o caminho relativo ou absoluto para o arquivo CSS. Por exemplo, *href="estilos.css"*.

Listas

Em HTML, existem três tipos principais de listas que podem ser criadas:

Listas Não Ordenadas ():

Uma lista não ordenada é uma lista de itens que não segue uma ordem específica.

Cada item da lista é precedido por um marcador de lista, geralmente um ponto preto.

Os elementos de uma lista não ordenada são definidos com a tag (item de lista).

Listas Ordenadas ():

Uma lista ordenada é uma lista de itens que segue uma ordem específica, geralmente numerada ou com marcadores.

Cada item da lista é numerado sequencialmente ou precedido por um marcador específico.

Os elementos de uma lista ordenada também são definidos com a tag .

E você pode colocar argumentos para criar novas metodologias de listagem

```

</ul>
<ol type="A">
  <li>Morango</li>
  <li>Tomate
    <ol type="i">
      <li>Vermelho</li>
      <li>Verde</li>
    </ol>
  </li>
</ol>

```

Listas de Definição(<dl>, <dt>, <dd>):

Uma lista de definição é uma lista que consiste em termos (definidos com <dt>) e suas definições (definidas com <dd>).

Cada termo é seguido por sua definição correspondente.

<dl> é a tag de lista de definição, <dt> é a tag para termos e <dd> é a tag para descrições dos termos.

```

<h2>Lista de descrição</h2>
<dl>
  <dt>Abacaxi</dt>
  <dd>É uma fruta bastante ácida...</dd>
  <dt>Limão</dt>
  <dd>É uma fruta bem azeda</dd>
</dl>

```

Pesquisa Google

Uma boa pesquisa no google deve conter :

- **<title>**: O título da página deve ser descritivo e incluir as palavras-chave relevantes para o conteúdo da página.
- **<meta name="description" content="...">**: A meta descrição deve ser uma breve descrição do conteúdo da página, incluindo as palavras-chave relevantes. Isso será exibido nos resultados de pesquisa do Google.
- **<meta name="keywords" content="...">**: As palavras-chave devem ser uma lista separada por vírgula de palavras ou frases que descrevam o conteúdo da página. No entanto, é importante notar que o Google não usa mais essa meta tag para indexar o conteúdo da página.
- **<link rel="canonical" href="...">**: A meta tag canônica é usada para indicar a versão preferida de uma página quando houver várias URLs que apontam para o mesmo conteúdo. Isso ajuda a evitar conteúdo duplicado e a concentrar a autoridade da página em uma única URL.

Comandos

<code><!-- --></code>	Define um comentário
<code><!DOCTYPE></code>	Define o tipo de documento
<code><a></code>	Define um hiperlink
<code><article></code>	Define um artigo
<code><aside></code>	Define o conteúdo, além do conteúdo da página
<code><audio></code>	Define o conteúdo de som incorporado
<code></code>	Define texto em negrito
<code><body></code>	Define o corpo do documento
<code><button></code>	Define um botão clicável
<code><div></code>	Define uma seção em um documento
<code><footer></code>	Define um rodapé para um documento ou seção
<code><form></code>	Define um formulário HTML para entrada do usuário
<code><h1> até <h6></code>	Define cabeçalhos HTML
<code><head></code>	Contém metadados / informações para o documento
<code><header></code>	Define um cabeçalho para um documento ou seção
<code><hr></code>	Define uma mudança temática no conteúdo
<code><html></code>	Define a raiz de um documento HTML
<code><iframe></code>	Define um quadro embutido
<code></code>	Define uma imagem
<code><input></code>	Define um controle de entrada
<code><label></code>	Define um rótulo para um elemento <code><input></code>
<code></code>	Define uma lista ordenada
<code></code>	Define uma lista não ordenada

`<!-- comentário-->`

Tag para se fazer comentários em um código HTML

`<!DOCTYPE html>`

Tag que define para o sistema que a partir dali temos um código html

`<html lang = “pt-BR”>`

Tag que define a linguagem do sistema, nesse caso, **pt-BR** pode ser substituído por outras línguas.

`<head> </head>`

Tags que respectivamente indicam o início e o final de um cabeçalho.

<meta charset = “UTF-8”>

Tag que define o tipo de caracteres a serem usados no site, para permitir que ele assimile caracteres de diferentes linguagens do mundo todo.

<meta name = “tipoDeMeta” content = “valorDoMeta”>

Tag base para se criar uma tag meta, veja a aba de <meta> para ver os diferentes tipos de valores que podem ser inseridos e uma tag meta.

<title> NomeDoTítulo </title>

Tags para definir o título de uma página.

<body> corpoDoSite </body>

Tags para definir o corpo de um site.

<hn> Palavras </hn>

Tags para se definir tamanhos de cabeçalho, a letra n pode ser substituída por números de **1 a 6**, sendo 1 o maior cabeçalho de todos, e 6 o menor.

<p> Lorem Ipsum </p>

Tags para se definir um parágrafo de texto em html.

<div> divisão </div>

Tags para se definir uma divisão em um site.

<script> scriptX </script>

Tags para se definir uma inserção de um script em html, o padrão são scripts em JavaScript.

<script type=“tipoDeScript”> scriptX </script>

Tags para se definir uma inserção de um script em html, quando se precisa definir um tipo de Script diferente de JavaScript.

```
window.onload = function (){  
  comando  
}
```

Comando em JavaScript para se executar um código JavaScript assim que os elementos da página estiverem carregados.

<script src= “arquivoJavaScript.js” type= “text/JavaScript”>

Tag para se implementar um arquivo de script, nesse caso, na linguagem JavaScript.

<style type = “text/Css”>

Tag para se adicionar Css em uma página html.

<link rel="stylesheet" type="text/css" href="ArqCSS.css">

Tag para se implementar um arquivo de css em um index html.

** **

Tags para se criar uma lista não ordenada em html.

** **

Tags para se criar itens de lista

** **

Tags para se criar uma lista ordenada em html.

<dl> </dl>

Tags para se criar uma lista de definição

<dt> </dt>

Tags para se criar os termos de uma lista de definições.

<dd> </dd>

Tags para se criar as descrições dos termos de uma lista de definições.

Ctrl + “

Atalho para abrir o terminal no VsCode.

<iframe atributos> </iframe>

Tag para se incorporar um “frame” ou pedaço de um site em nosso próprio site.

** palavraDeLink **

Tag de âncora, que te permite ancorar um link. href indica a referência de que link você quer mandar o usuário. palavraDeLink se refere a palavra que você deseja que seja clicável.

** palavraDeLink **

Adaptação da tag de âncora, para que ao clicar nesse link, o usuário seja direcionado a uma nova página. palavraDeLink se refere a palavra que você deseja que seja clicável.

<form> </form>

Tag para criação de formulário

<label for=nome> Nomezinho: </label>

Tag para criação de uma label.

Nesse caso, o identificador da label é ‘nome’, e o que aparecerá na tela é ‘Nomezinho:’ .

<input type = "text" id = "nome">

Tag para dar input em uma informação em uma label

Nesse caso, **'text'** se refere ao tipo de dado que vai ser inserido ali, e **'nome'** se refere ao identificador da label que colocamos na tag label acima..

<textarea> </textarea>

Tag para se criar uma caixa de texto de tamanho determinável.

<button> nomeBotao </button>

Tag para se criar um botão.

Links e afins

Cookbook W3Schools, tutorial HTML, inglês:

<https://www.w3schools.com/html/default.asp>

Dúvidas