

# CURSO DE FRONT-END

Aprendendo Front-end desenvolvendo jogos para web



**Professor/autor:**  
Leonardo S. Souza



# PREVIEW DO CURSO

## 1- HTML 5

Neste módulo, vamos aprender as principais funções do HTML, que incluem desde a adição de elementos visíveis para o usuário, como textos, imagens e ícones, até a inserção de metadados.

## 2- CSS 3

Após aprendermos sobre as principais funcionalidades do HTML, iremos utilizar os elementos inseridos para definirmos cor, tamanho, posição e outras diversas estilizações presentes no CSS, com o intuito de deixar a página esteticamente bonita.

## 3- JAVASCRIPT

Com o JavaScript, poderemos definir a lógica dentro de uma página, sendo responsável por criar interatividade com o usuário. Aprenderemos desde os conceitos básicos de programação com essa linguagem até a manipulação do DOM com JavaScript.

## 4- CONSTRUINDO PROJETOS

Para finalizarmos, construiremos juntos mini projetos. Neles iremos desenvolver dois jogos para web, utilizando as linguagens aprendidas anteriormente.

# MÓDULOS 1 - HTML 5



# SUMÁRIO

**1- O QUE É O HTML?**

**2-ESTRUTURA BÁSICA**

**3- COMENTÁRIOS**

**4- ELEMENTOS COMUNS NO HTML**

**5- HIERARQUIA DE TÍTULOS H1 A H6**

**6- PARÁGRAFOS**

**7- IMAGENS**

**8- LINKS**

**9- LISTA ORDENADA**

**10-LISTA NÃO ORDENADA**

**11- FORMULÁRIO**

**12- USO DE CLASSE E ID**

**13- CONTAINERS HTML**

**14- CONTAINER GENÉRICO**

**15- HIERARQUIA DE ELEMENTOS E  
INDENTAÇÃO NO HTML**

**16- CONTAINER SEMÂNTICO**

# O QUE É O HTML?

O hypertext markup language ou HTML, é uma linguagem de marcação que tem como principal objetivo a manipulação de elementos visíveis ao usuário, ou seja, todos os títulos, parágrafos, imagens, vídeos e ícones vistos na web diariamente, são elementos marcados pela mesma. As tags HTML são declaradas por meio do sinal de menor e maior na abertura e sinal de menor, barra e sinal de maior no fechamento, da seguinte maneira:

**<ABERTURA></FECHAMENTO>**

# ESTRUTURA BÁSICA

```
<!DOCTYPE HTML>  
<HTML LANG="PT-BR">  
  <HEAD>  
  
    <META CHARSET="UTF-8">  
    <TITLE>NOME DA PAGINA</TITLE>  
  
  </HEAD>  
  <BODY>  
  
  </BODY>  
</HTML>
```

```
<!DOCTYPE HTML> <!--INDICA AO NAVEGADOR O TIPO DE TEXTO-->
```

Indica para o navegador que este arquivo é do tipo HTML, **indicando também a sua versão.**

```
<HTML LANG="PT-BR"> </HTML>
```

Esta tag **engloba todas as outras tags HTML** que virão posteriormente. e o atributo **lang**, define qual o idioma predominante da página, como o exemplo mostra, o idioma predominante é o **"pt-br"**, ou seja, português-brasil.

```
<HEAD> </HEAD>
```

Esta tag **receberá todos os dados que não serão visíveis ao usuário**, sendo elas informações do documento que serão interpretadas pelo navegador (**Metadados**), como: **links para folha de estilo CSS; Scripts; título do documento.**

```
<META CHARSET="UTF-8"> <!-- TIPOS DE CARECTERES -->
```

Este metadato definirá os tipos de caracteres que serão lidos neste documento, o UTF-8 é o mais comum na web.

```
<TITLE> </TITLE> <!-- TITULO -->
```

Esta tag define o título da página.

```
<BODY> </BODY> <!-- TAG DO CORPO DA PAGINA -->
```

Sendo o corpo da página, esta tag receberá todos os elementos que serão visíveis para o usuário, como: textos; imagens; vídeos; ícones.



# COMENTARIOS

O comentário é uma ferramenta de boa prática para que você se lembre do que estava fazendo e para deixar claro para um parceiro de trabalho o que determinado código é responsável. Os comentários no HTML são declarados da seguinte maneira:

```
<!-- COMENTARIO -->
```

## ATALHO NO VS CODE

CTRL

+

;

# ELEMENTOS COMUNS NO HTML

Dentro do HTML, existem vários tipos de tags. No entanto, algumas são de uso mais comum, associadas a elementos como **títulos, parágrafos, imagens, botões, links, listas e containers**, que são observadas diariamente pelo programador Front-End.

Títulos - `<h1>` à `<h6>`

```
<H1></H1>
```

Parágrafos

```
<p></p>
```

Imagem

```
<IMG SRC=""></IMG>
```

Botão

```
<BUTTON></BUTTON>
```

Links

```
<a></a>
```

Lista ordenada

```
<ol></ol>
```

Lista não ordenada

```
<ul></ul>
```

Itens de uma lista

```
<li></li>
```

Container genérico

```
<DIV></DIV>
```

# HIERARQUIA DE TITULOS H1 A H6

Os títulos em HTML são organizados em ordem decrescente de importância. O `<h1>` é o título mais importante da página, enquanto o `<h6>` é o menos importante. Essa organização segue uma hierarquia: o `<h1>` é o **título principal**, o `<h2>` é o **subtítulo** do `<h1>`, o `<h3>` é **subtítulo** do `<h2>`, e assim por diante, até chegar ao `<h6>`, que representa o nível mais baixo na hierarquia de títulos.

```
<BODY>
```

```
  <H1>TITULO PRINCIPAL</H1>
```

```
  <H2>SUB-TITULO</H2>
```

```
  ...
```

```
</BODY>
```

# PARAGRAFOS

A tag `<p>` é responsável por definir parágrafos em uma página web, ajudando na **organização e na separação dos blocos de texto**.

```
<P>PRIMEIRO PARAGRAFO</P>  
<P>SEGUNDO PARAGRAFO</P>  
<P>TERCEIRO PARAGRAFO</P>
```

# IMAGENS

A tag `<img>` é **responsável pela inserção de imagens** em uma página web. O atributo `src` é o principal responsável por isso, pois é nele que **inserimos o link ou o arquivo da imagem** que desejamos exibir. Já o atributo `alt` **serve como uma alternativa para identificar a imagem caso ela não seja carregada**. Além disso, é uma importante ferramenta de acessibilidade, pois permite que leitores de tela informem ao usuário que há uma imagem naquela posição e descrevam seu conteúdo.

```
<IMG SRC="LINK_DA_IMAGEM" ALT="TEXTO_ALTERNATIVO">
```

# LINKS

A tag `<a>` possui um papel fundamental, pois é **responsável por toda a navegação na web**, direcionando os usuários para outras páginas, seções ou arquivos dentro do mesmo site. É no atributo `href` que **inserimos o link responsável por esse direcionamento**. Já o atributo `target` **define onde o link será aberto**, como por exemplo em uma nova aba do navegador.

```
<A HREF="URL" ALT="TEXTO ALTERNATIVO PARA IMAGEM">LINK</A>
```

# VALORES DO ATRIBUTO TARGET :

- **\_blank** – Abre o documento vinculado em uma nova janela ou aba.
- **\_self** – Abre o documento vinculado na mesma janela ou aba em que foi clicado (padrão).
- **\_parent** – Abre o documento vinculado no frame pai.
- **\_top** – Abre o documento vinculado no corpo completo da janela.
- **framename** – Abre o documento vinculado no iframe com o nome especificado.

# LISTA ORDENADA

A tag `<ol>` é responsável por criar uma lista com itens ordenados de **forma sequencial**, sendo marcada por números, letras, números romanos, entre outros.

```
<OL> <!-- LISTA ORDENADA -->
```

```
<LI>ITEM</LI>
```

```
<LI>ITEM</LI>
```

```
<LI>ITEM</LI>
```

```
<LI>ITEM</LI>
```

```
<!-- A TAG <LI> É RESPONSÁVEL POR REPRESENTAR UM ITEM DENTRO DE  
UMA LISTA, SEJA ELA ORDENADA OU NÃO. -->
```

```
</OL>
```



# LISTA NÃO ORDENADA

A tag `<ul>` é **responsável por criar uma lista de itens sem uma ordem definida**. Os itens dessa lista são marcados por símbolos, sendo o padrão um círculo preenchido.

```
<ul> <!-- LISTA NÃO ORDENADA -->
```

```
<li>ITEM</li>
```

```
<li>ITEM</li>
```

```
<li>ITEM</li>
```

```
<li>ITEM</li>
```

```
<!-- A TAG <li> É RESPONSÁVEL POR REPRESENTAR UM ITEM DENTRO DE UMA  
LISTA, SEJA ELA ORDENADA OU NÃO. -->
```

```
</ul>
```

# FORMULÁRIO

Um **formulário HTML** é uma estrutura que **permite ao usuário interagir com a página e inserir informações**. Ele é composto por diversos **elementos de entrada** (também chamados de inputs), cada um com uma função específica. Veja os principais:

- **Campos de texto:** usados para inserir textos simples, como nomes ou endereços.
- **Campos de e-mail:** validados automaticamente pelo navegador para garantir que o formato do e-mail esteja correto.
- **Campos de senha:** ocultam os caracteres digitados para proteger informações sensíveis.
- **Áreas de texto (textarea):** ideais para textos mais longos, como comentários ou descrições.
- **Caixas de seleção (checkboxes):** permitem marcar múltiplas opções de uma lista.
- **Botões de opção (radio buttons):** usados para selecionar apenas uma opção entre várias.
- **Listas suspensas (select):** exibem uma lista compacta, permitindo a seleção de uma opção.
- **Botões (button ou input type="submit"):** usados para enviar o formulário ou executar outras ações.

## TAG – FORM:

- Define o **início de um formulário**.
- Os atributos **action** e **method** indicam:
  - **action:** para onde os dados serão enviados (URL ou endpoint).
  - **method:** o método **HTTP** usado, como post ou get.

## TAG - LABEL:

- Representa o **rótulo descritivo** de um campo.
- O atributo **for** deve ter o mesmo valor do **id** do campo correspondente.

# TAG - INPUT:

- Elemento genérico de entrada de dados. O tipo de dado é definido com o atributo **type**.
- **Exemplos:**
  - **type="text"**: campo de texto simples.
  - **type="email"**: valida se o valor inserido é um e-mail válido.
  - **type="password"**: esconde os caracteres digitados.
  - **type="checkbox"**: cria caixas de seleção.
  - **type="radio"**: cria botões de opção (somente um pode ser selecionado por grupo).
  - **type="submit"**: botão para enviar o formulário (não foi usado no exemplo, usamos **<button>** no lugar).

## TAG - TEXTAREA

- Usado para **campos de texto maiores**, como comentários ou descrições.

## TAG - SELECT

- Cria uma **lista suspensa** de opções.

## TAG - OPTION

- Define cada **item disponível** em uma **<select>**.

# TAG - BUTTON

- Representa um **botão clicável**.
- Com `type="submit"`, ele envia o formulário.
- Pode também ser `type="button"` (**para ações com JS**) ou `type="reset"` (**para limpar o formulário**).

# USO DE CLASSE E ID

## CLASSE :

As **classes** são atributos utilizados em elementos HTML para facilitar a **padronização e a organização do código**. Elas são muito úteis para aplicar estilos **CSS** e manipular elementos via **JavaScript**, tornando o desenvolvimento web mais eficiente. Uma classe é definida no atributo **class** dentro de uma tag HTML. **O nome da classe pode ser qualquer palavra sem espaços e caracteres especiais (exceto - e \_), sendo obrigatório começar com uma letra.**

```
<P CLASS="NOME_DA_CLASSE" > </P>
```



## CASOS DE USO DA CLASSE :

- ✓ Aplicar o mesmo estilo a vários elementos.
- ✓ Criar padrões visuais reutilizáveis no CSS.
- ✓ Manipular vários elementos ao mesmo tempo com JavaScript.
- ✓ Melhor organização do código.

# ID:

○ **ID** é um atributo utilizado nos elementos HTML para identificá-los de forma única dentro da página web. Diferente das classes, um **ID** deve ser exclusivo, ou seja, **cada elemento pode ter apenas um ID**, e esse **ID não pode ser repetido em outro elemento** na mesma página.

```
<P ID="NOME_DO_ID" > </P>
```

## CASOS DE USO DO ID :

- ✓ Criar âncoras (links internos que levam a uma seção específica da página).
- ✓ Manipular um elemento específico com JavaScript.
- ✓ Estilizar um elemento único na página com CSS.
- ✓ Aplicar regras de CSS mais fortes (pois IDs têm maior prioridade que classes).

# CONTAINERS HTML

Os **containers** são elementos responsáveis por agrupar e organizar outros elementos com uma finalidade específica, como uma caixa que armazena diversos itens — assim como um container de caminhão, que armazena e organiza produtos para transportá-los de um lugar para o outro. No HTML, a criação de **um container não tem função visual por si só, mas é fundamental para a criação de layouts e para a divisão de elementos**. As principais finalidades de um container são: organizar o código, aplicar estilos CSS em determinados blocos de código, criar layouts responsivos (posicionando os elementos com **Flexbox** ou **Grid**) e manipular elementos específicos com **JavaScript**.

# CONTAINER GENÉRICO

A tag `<div>` é responsável por criar um container genérico, sem nenhum significado semântico, sendo usada para organizar diversos elementos.

```
<D<div> <!-- CONTAINER GENÉRICO -->


// OUTROS ELEMENTOS ...

</div>
```




# HIERARQUIA DE ELEMENTOS E INDENTAÇÃO NO HTML

A **hierarquia de elementos dentro do HTML é definida pela forma como os containers são estruturados**. Podemos comparar essa hierarquia com uma **árvore genealógica**, onde um elemento pai pode conter múltiplos filhos, e cada um desses filhos pode conter seus próprios subelementos.






## HIERARQUIA DE ELEMENTOS (PAI, FILHO E IRMÃO):

- **Elemento Pai:** Um container que envolve outros elementos dentro dele.
  - **Elemento Filho:** Elementos que estão dentro de um container pai.
  - **Elemento Irmão:** Elementos que estão no mesmo nível hierárquico, ou seja, compartilham o mesmo pai.
- 



## INDENTAÇÃO NO HTML:

**A indentação no HTML é uma prática essencial para manter o código organizado e facilitar a leitura.** Os elementos descendentes de um container, seus filhos, terão um recuo para direita, e os elementos que são irmãos, se mantêm, no mesmo nível. podemos observar melhor no exemplo a seguir:





# EXEMPLO:

```
<!--ELEMENTO PAI-->
  <DIV CLASS="CONTAINER_PAI">
    <!--ELEMENTO FILHO DO CONTAINER_PAI-->
    <DIV CLASS="FILHO_1">
      <!--ELEMENTO FILHO DE FILHO_1 -->
      <P></P>
    </DIV>

    <!--FILHO_1 E FILHO_2 SÃO IRMÃOS, ELEMENTOS DE MESMO NÍVEL
    HIERARQUICO -->

    <!--ELEMENTO FILHO DO CONTAINER_PAI-->
    <DIV CLASS="FILHO_2">
      <!--ELEMENTO FILHO DE FILHO_2 -->
      <IMG SRC="URL" ALT="TEXTO_ALTERNATIVO">
    </DIV>
  </DIV>
```

# CONTAINERS SEMÂNTICOS

Os **containers semânticos** no HTML5 são fundamentais para melhorar a acessibilidade e a otimização para mecanismos de busca (SEO). Eles tornam o **código mais organizado, facilitam a interpretação pelos leitores de tela e ajudam os motores de busca** a entenderem melhor a estrutura da página. Além disso esses containers **substituem o uso excessivo de `<div>`**, tornando o código mais legível e estruturado.

# TAGS DOS CONTAINERS SEMANTICOS :

**<header></header>** <!-- Cabeçalho do site ou seção -->

**<nav></nav>** <!-- Menu de navegação -->

**<main></main>** <!-- Conteúdo principal -->

**<section></section>** <!-- Seção independente dentro do site -->

**<article></article>** <!-- Conteúdo autônomo (ex: post de blog) -->

**<aside></aside>** <!-- Conteúdo secundário (ex: barra lateral) -->

**<footer></footer>** <!-- Rodapé do site -->

# FIM DO MÓDULO !

“Na vida, não existe nada a temer, mas a entender.”

Marie Curie