# **Conceitos Fundamentais**

# O que é o HTML?

- HTML significa Hyper Text Markup Language;
- HTML é a linguagem de marcação padrão para criação de páginas da Web;
- HTML descreve a estrutura de uma página da Web;
- HTML consiste em uma série de elementos;
- Os elementos HTML informam ao navegador como exibir o conteúdo.

# O que é um elemento HTML?

Um elemento HTML é definido por uma tag inicial, algum conteúdo e uma tag final:

< tagname > O conteúdo vai aqui... < /tagname > Tudo isso é um elemento!

## Principais tags utilizadas:

# **ESTRUTURA DO DOCUMENTO:**

br html>	Auto- fechante	Define o tipo do documento
<html></html>		Elemento raiz de um documento HTML
<head></head>		Contém metadados e links
<body></body>		Contém o conteúdo visível

# **METADADOS E LINKS:**

<title>&lt;/th&gt;&lt;th&gt;</title>	Define o título da página	
<meta/>	Auto- fechante	Metadados (charset e viewport)
<li>k&gt;</li>	Auto- fechante	Relação entre o documento e um recurso externo
<script></th><th></script>	Inclui um script JavaScript.	
<style></th><th></style>	Contém CSS interno	
<a></a>		Link/ âncora

# **ESTRUTURA E LAYOUT:**

<header></header>		Cabeçalho da página
<nav></nav>		Navegação
<main></main>		Conteúdo principal
<footer></footer>		Rodapé da página
<div></div>		Divisão genérica
<span></span>		Texto em linha

# **TEXTO E FORMATAÇÃO:**

<h1></h1>		Cabeçalho de nível 1
		Parágrafo
<strong></strong>		Texto em negrito (ênfase)
<em></em>		Define texto em itálico (ênfase)
<b></b>		Texto em negrito (sem ênfase)
<i>&gt;</i>		Texto em itálico (sem ênfase)
-		

<code></code>		Define texto como código
<q></q>		Define uma abreviação ou acrônimo.
<address></address>		Define informações de contato

#### LISTAS:

<ul><li><ul></ul></li></ul>		Lista não ordenada
<ol></ol>		Lista ordenada
<li><li>&lt;</li></li>		Item da lista

#### TABELAS:

	Tabela
	Linha da tabela
>	Célula de dados
	Célula de cabeçalho

# FORMUL ÁRIOS:

<form></form>		Formulário
<input/>	Auto-fechante	Campo de entrada
<textarea>&lt;/th&gt;&lt;th&gt;</textarea>	Área de texto	
<button></button>		Botão
<select></select>		Menu suspenso
<option></option>		Opção no menu
<label></label>		Rótulo para o campo
<fieldset></fieldset>		Grupo de campos
<legend></legend>		Legenda para o grupo

# **ELEMENTOS DE LINHA E SEPARADORES:**

	Auto-fechante	Insere uma quebra de linha
<hr/>	Auto-fechante	Insere uma linha horizontal

#### MÍDIA:

<img/>	Auto-fechante	Imagem
--------	---------------	--------

#### **Atributos**

São pares de nome-valor usados para fornecer informações adicionais sobre elementos HTML. Eles são colocados dentro das tags de abertura dos elementos e ajudam a definir propriedades ou comportamentos específicos para esses elementos.

<a href="https://example.com">Visite o site</a>
<img src="image.jpg" alt="Descrição da Imagem">
<div class="container">Conteúdo</div>

href, src, alt e class são ATRIBUTOS

#### O que é o CSS?

- CSS significa Cascading Style Sheets:
- CSS descreve como os elementos HTML devem ser exibidos na tela.
- CSS pode controlar o layout de várias páginas da web de uma só vez
- Folhas de estilo externas são armazenadas em arquivos CSS

#### **Sintaxe CSS:**



Seletor: elemento HTML que deseja estilizar.

Declaração: nome da propriedade CSS + valor (separados por :)

## Seletores CSS SELETOR DE ID:

O **id** de um elemento **é único** dentro de uma página, então o seletor id é usado para selecionar um elemento único! Para selecioná-lo escreva: **#**id

```
#para1 {
   text-align: center;
   color: red;
}

Mota: um nome de ID não
pode começar com um
número!
}
```

# **SELETOR DE CLASSE:**

O seletor de classe seleciona elementos HTML com um atributo de classe específico.

Para selecioná-lo escreva: .classe

```
.center {
  text-align: center;
  color: red;
  color: red;
}
```

SELETOR DE ELEMENTO	SELETOR AGRUPAMENTO
<pre>h1 {    text-align: center;    color: red; }</pre>	<pre>h1, h2, p {    text-align: center;    color: red; }</pre>

SELETOR UNIVERSAL	
<pre>* {   text-align: center;   color: blue; }</pre>	

## **Bordas**

As propriedades de borda CSS permitem que você especifique o estilo, a largura e a cor da borda de um elemento.

# Propriedade: border-style

#### Valores:

- dotted- Define uma borda pontilhada
- dashed- Define uma borda traceiada
- solid- Define uma borda sólida
- double- Define uma borda dupla
- groove- Define uma borda ranhurada 3D.
- ridge- Define uma borda 3D com sulcos.

- inset- Define uma borda de inserção 3D.
- outset- Define uma borda de início 3D.
- none- Não define nenhuma fronteira
- hidden- Define uma borda oculta

#### Propriedade: border-width

<u>Valores</u>: A largura pode ser definida como um tamanho específico (em **px**, **pt**, **cm**, **em**, etc.) ou usando um dos três valores predefinidos: **thin**, **medium**, ou **thick**.

#### Propriedade: border-color

#### Valores:

- nome especifique um nome de cor, como "red"
- HEX valor HEX, como "#ff0000"
- RGB valor RGB, como "rgb(255,0,0)"
- HSL valor HSL, como "hsl(0, 100%, 50%)"
- transparent gsg

**Nota:** Se o border-color não for passado ele herda a cor do elemento.

#### Propriedade: border-radius

<u>Valores</u>: A largura pode ser definida como um tamanho específico (em **px**, **pt**, **cm**, **em**, etc.).

#### Margens

Margens são usadas para criar espaço ao redor de elementos, fora de quaisquer bordas definidas.

#### Propriedades:

- margin-top
- margin-right
- margin-bottom
- margin-left
- margin

#### Valores:

- auto o navegador calcula a margem
- comprimento margem em px, pt, cm, etc.
- % % da largura do elemento que contém
- inherit- margem deve ser herdada do elemento pai

#### Posição

A propriedade position define como um elemento é posicionado na página:

- Static: Posição padrão, fluxo normal do documento.
- Relative: Posição ajustada em relação à posição original do elemento, utilizando top, right, bottom, e left.
- **Absolute**: Posição em relação ao ancestral mais próximo que não seja static.
- **Fixed:** Posição em relação à janela de visualização, permanecendo fixa durante o rolar da página.

## **Grid Layout**

Grid Layout é um modelo de layout CSS que permite criar layouts bidimensionais, organizando os itens em linhas e colunas. As principais propriedades são:

- **Display: grid**: Define um contêiner de grade.
- **Grid-template-rows** e **Grid-template-columns**: Define o tamanho das linhas e colunas da grade.
- **Grid-area**: Especifica a área em que um item deve ser colocado.
- **Grid-column** e **Grid-row**: Controlam a posição e o tamanho dos itens dentro da grade.
- Gap: Define o espaço entre as linhas e colunas da grade.

#### **Padding**

O Padding é usado para criar espaço ao redor do conteúdo de um elemento, dentro de quaisquer bordas definidas.

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
```

#### **Display**

A propriedade "display" controla como os elementos são

exibidos e como interagem com layout.

Block	Ocupa toda a largura disponível e inicia uma nova linha.
Inline	Ocupa apenas o espaço necessário e não inicia uma nova linha.
Inline-block	Se comporta como inline, mas permite definir largura e altura, e comporta-se como um bloco em termos de layout.

#### **Display Flex**

O display: flex é um modelo de layout que facilita a criação de layouts flexíveis e responsivos. As principais propriedades incluem:

- **Flex-direction:** Define a direção dos itens (horizontal ou vertical).
- Flex-wrap: Controla se os itens devem quebrar em várias linhas.
- **Flex-grow:** Define a capacidade de um item crescer para ocupar espaço extra.
- Flex-shrink: Define a capacidade de um item encolher quando necessário.
- **Flex-basis:** Define o tamanho inicial de um item antes da distribuição do espaço.
- **Justify-content:** Alinha os itens ao longo do eixo principal (horizontal).
- **Align-items:** Alinha os itens ao longo do eixo transversal (vertical).
- Align-content: Alinha as linhas de itens quando há múltiplas linhas.
- **Align-self:** Alinha um item individualmente dentro do contêiner flexível.

#### O que é o CSS?

- Interatividade Web: JavaScript cria interatividade em páginas web.
- **Execução no Navegador:** Rodado no navegador para atualizar páginas sem recarregar (lado do cliente).
- Manipulação do DOM: Manipulação e alteração da estrutura HTML e CSS das páginas web.

Nota: O DOM é uma estrutura que representa a página web como uma árvore de elementos, permitindo a manipulação do conteúdo e estrutura da página com JavaScript.)

# Sintaxe Básica JavaScript

## VARIÁVEIS:

- O que são: São como caixas para guardar dados na memória do seu computador.
- Para que servem: Armazenam informações que seu programa irá usar.
- **Como declarar:** Usamos palavras-chave como var, let ou const para criar uma variável.
- Tipos:
  - o var: Permite redeclarar e reatribuir.

- let: Permite redeclarar, mas não reatribuir a uma constante.
- const: Cria uma constante, seu valor não muda.
- Escopo: O lugar onde a variável é válida (global, local)

#### **TIPOS DE DADOS:**

- Números: (Number) Para valores numéricos;
- Strings: Para textos (ex: "Olá, mundo!);
- Booleanos: Verdadeiro (true) ou falso (false);
- Null: Indica a ausência intencional de um valor;
- **Undefined:** Indica que a variável foi declarada mas ainda não tem valor definido.
- Arrays: Listas de valores (ex: ["maçã", "morango"])
- **Objetos:** Conjuntos de chave-valor (ex: { nome: "João", idade: 30 }).

#### **OPERADORES:**

São símbolos que realizam operações em variáveis e valores. Em JavaScript, eles são classificados em três tipos principais:

- Aritméticos: Realizam operações matemáticas básicas.
  - + (adição)
  - (subtração)
  - \* (multiplicação)
  - / (divisão)
- 2. **Relacionais**: Comparam valores e retornam um resultado booleano.
  - == (igual a) ignorando o tipo.
  - === (estritamente igual a) inclusive tipos.
  - o > (maior que)
  - < (menor que)</p>
- 3. **Lógicos**: Realizam operações lógicas entre valores booleanos.
  - && (E): Verdadeiro se ambas as condições forem verdadeiras.
  - | (OU): Verdadeiro se pelo menos uma das condições for verdadeira.
  - ! (NÃO): Inverte o valor booleano, retornando verdadeiro se a condição for falsa.

# **FUNÇÕES:**

Blocos de código function myFunction(a, b) {
reutilizáveis que realizam
uma tarefa específica.
}

# ESCOPO:

Refere-se à visibilidade das variáveis. Variáveis declaradas dentro de uma função têm escopo local, enquanto variáveis declaradas fora têm escopo global.

#### **CONTROLE DE FLUXO E LOOPS:**

Permite que você tome decisões e execute diferentes blocos de código dependendo das condições.

# Condicionais:

- **if**: Executa um bloco de código se a condição for verdadeira.
- else: Executa um bloco de código alternativo se a condição if for falsa.
- else if: Adiciona condições adicionais para testar.

**Loops:** Permitem executar um bloco de código repetidamente.

- for: Itera sobre um bloco de código um número específico de vezes.
- while: Executa um bloco de código enquanto a condição for verdadeira.

#### Manipulação do DOM

# **SELECIONANDO ELEMENTOS:**

- getElementById: Seleciona um elemento pelo ID.
- querySelector: Seleciona o primeiro elemento que corresponde a um seletor CSS.

```
<div id="example1">Texto do elemento com ID 'example1'</div>
<div class="example2">Texto do elemento com classe 'example2'</div>
<button onclick="changeContent()">Mudar Conteúdo</button>
<script>
    function changeContent() {
        // Usando getElementById
        var elementById = document.getElementById('example1');
        elementById.textContent = 'Conteúdo alterado pelo getElementById';
        elementById.style.color = 'green'; // Alterando a cor do texto

        // Usando querySelector
        var elementByClass = document.querySelector('.example2');
        elementByClass.textContent = 'Conteúdo alterado pelo querySelector';
        elementByClass.style.fontSize = '25px'; // Alterando o tamanho da fonte
}
```

#### **EVENTOS:**

São ações que ocorrem em uma página web, como cliques ou pressionamentos de tecla, que podem ser detectadas e manipuladas usando JavaScript.

#### Click

É disparado quando um usuário clica com o botão esquerdo do mouse em um elemento.

#### <u>Mouseover</u>

Ocorre quando o cursor do mouse passa sobre um elemento.

#### Keydown

Ocorre quando uma tecla é pressionada enquanto um elemento tem o foco.

```
document.addEventListener('keydown', function (event) {
    if (event.key === 'Enter') {
        alert('Tecla Enter pressionada!');
    }
});
```

## **Onclick**

É acionado quando um usuário clica com o mouse em um elemento HTML.

```
<button onclick="alert('Botão clicado!')">
     Clique em Mim
  </button>
```

# DICAS:

# Atalhos interessantes no Visual Studio Code!

Alt + Shift + F	Formatar identação do código
Control + S	Salvar documento
Control + D	Seleciona elementos iguais
Shift + 1 Enter	Escreve documento HTML base
F2	Ajuda a reescrever a tag
Control + Z	Desfazer
Control + Shift + Z	Refazer
Control + ;	Comentar código
Control + F	Pesquisar palavra em específico

# Instalar Extensão Live Server no Visual Studio Code:



## Preste Atenção!

Para caminho de arquivos:

	Sair da pasta de onde você está digitando
/	Acessar pasta

#### **EXEMPLO:**

```
<img src="./Img/FotoPerfil.png" >
```

# Lembre-se: ANEXAR SUA FOLHA DE ESTILO (CSS) NO HTML EXEMPLO:

<link rel="stylesheet" href="./CSS/styles.css">

# Lembre- se: SEMPRE FECHAR AS TAGS

#### Sites de Apoio:

- https://www.w3schools.com
- Color Picker
- Google Fonts: Browse Fonts

