

HTML & JavaScript para Iniciantes

Guia pratico para alunos do SENAI
Aprenda a estrutura da web do zero!

Programação de Aplicativos | Nivel Iniciante | Curso Técnico SENAI

1. O que é HTML?

HTML significa **HyperText Markup Language** — Linguagem de Marcação de Hipertexto.

Ela diz ao navegador **o que mostrar** na tela: títulos, textos, imagens, botões, links...

Pense assim: se uma página web fosse uma casa, o HTML seria a **planta baixa** — ele define onde ficam as paredes, portas e janelas.

Analogia: os 3 pilares da web

HTML	CSS	JavaScript
Esqueleto / Estrutura	Aparência / Estilo	Comportamento / Interação
Define o que existe na página	Define como fica bonito	Define o que acontece

Estrutura básica de um arquivo HTML

index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Minha Primeira Página</title>
</head>
<body>
<h1>Ola, mundo!</h1>
<p>Este é o meu primeiro paragrafo.</p>
</body>
</html>
```

Regra de ouro: toda tag que abre precisa ser fechada. Exemplo: `<h1>` abre e `</h1>` fecha. A barra `/` antes do nome indica o fechamento.

2. As Tags mais Usadas

Tags são como **etiquetas** que embrulham o conteúdo e dizem o que ele é. Veja as principais:

Tag	Para que serve	Exemplo de uso
<h1> ate <h6>	Títulos (h1 maior, h6 menor)	<h1>Meu Título</h1>
<p>	Parágrafo de texto	<p>Um texto qualquer</p>
<a>	Link (âncora)	Clique aqui
	Inserir imagem	
 + 	Lista com marcadores	Item 1
 + 	Lista numerada	Primeiro
<div>	Caixa / agrupador genérico	<div>conteúdo aqui</div>
<button>	Botão clicável	<button>Clique!</button>
<input>	Campo para o usuário digitar	<input type="text">
	Destaque inline no texto	Texto especial
<form>	Formulario de dados	<form>...</form>

Dica: algumas tags não precisam ser fechadas porque não tem conteúdo interno.
Exemplos: , <input> e
 (quebra de linha).

3. Atributos: configurando as Tags

Atributos são como **configurações extras** colocadas dentro da tag de abertura. Eles dão mais informações sobre o elemento.

Formato: `nome-do-atributo="valor"`

Analogia: a tag é o carro. Os atributos são as configurações do carro: cor, tamanho, modelo. Sem atributos o carro existe, mas sem personalidade. Com atributos você define exatamente como ele vai aparentar!

Exemplo visual: a tag com seus atributos

Atributos na pratica

```
// Uma imagem com varios atributos:  
  
  
  
// Um link com atributos:  
  
<a href="https://senai.br" id="link-senai" target="_blank">  
  Visite o SENAI  
</a>
```

Atributos mais comuns

Atributo	Usado em	O que faz
src		Caminho ou URL da imagem
href	<a>	Endereço do link
id	qualquer tag	Identificador único do elemento (usado pelo JS)
class	qualquer tag	Grupo de estilo (usado pelo CSS e JS)
type	<input>	Tipo do campo: text, password, number, email...
alt		Texto descritivo caso a imagem não carregue
width/height		Largura e altura da imagem em pixels

placeholder	<input>	Texto de dica dentro do campo vazio
disabled	<button>	Desativa o elemento (nao pode ser clicado)
target	<a>	target="_blank" abre o link em nova aba

4. JavaScript: dando vida ao HTML

O **JavaScript (JS)** é a linguagem que faz a página **reagir**. Com ele você pode modificar o HTML em tempo real, sem recarregar a página!

Como funciona: o JS primeiro **encontra** um elemento HTML (pelo id ou classe), e depois **faz alguma coisa** com ele: muda o texto, muda a cor, esconde, mostra...

Como o JS encontra elementos HTML?

O elemento HTML precisa ter um `id` para o JS achar ele facilmente:

`getElementById` — o mais comum

```
// No HTML voce marca o elemento com um id:  
<h1 id="titulo-principal">Ola!</h1>  
  
// No JS voce usa getElementById para encontrar:  
let titulo = document.getElementById('titulo-principal');  
  
// E ai faz o que quiser com ele:  
titulo.textContent = 'Texto novo aqui!';  
titulo.style.color = 'blue';
```

Outra forma: `querySelector` (pelo nome da classe)

`querySelector`

```
// No HTML, o elemento tem uma class:  
<p class="destaque">Paragrafo importante</p>  
  
// No JS, usamos querySelector com ponto antes do nome:  
let paragrafo = document.querySelector('.destaque');  
paragrafo.style.backgroundColor = 'yellow';
```

Resumo rápido: `getElementById('nome')` busca pelo **id** (sem prefixo).
`querySelector('.nome')` busca pela **class** (com ponto na frente).

5. Reagindo a Cliques e Outras Ações

O JS consegue **escutar eventos** do usuário: cliques, digitação, passagem do mouse... Para isso usamos o `addEventListener`.

Evento de clique

```
// HTML:
<button id="meu-btn">Clique aqui!</button>
<p id="resposta">Aguardando...</p>

// JavaScript:
let btn = document.getElementById('meu-btn');
let resposta = document.getElementById('resposta');

btn.addEventListener('click', function() {
  resposta.textContent = 'Voce clicou no botao!';
  resposta.style.color = 'green';
});
```

O que você pode fazer com um elemento encontrado

O que fazer	Codigo JavaScript
Mudar o texto	<code>elemento.textContent = 'novo texto';</code>
Inserir HTML dentro	<code>elemento.innerHTML = 'negrito';</code>
Mudar cor do texto	<code>elemento.style.color = 'red';</code>
Mudar cor de fundo	<code>elemento.style.backgroundColor = 'yellow';</code>
Esconder o elemento	<code>elemento.style.display = 'none';</code>
Mostrar o elemento	<code>elemento.style.display = 'block';</code>
Adicionar classe CSS	<code>elemento.classList.add('destaque');</code>
Remover classe CSS	<code>elemento.classList.remove('destaque');</code>
Mudar um atributo	<code>elemento.setAttribute('src', 'nova.jpg');</code>
Ler o que o usuário digitou	<code>let valor = meuInput.value;</code>

Dica: o JS é executado de cima para baixo. Coloque sua tag `<script>` sempre antes do `</body>` para garantir que o HTML já carregou antes de o JS tentar encontrar os elementos.

6. Exemplo Completo: App de Saudacao

Veja um mini-app funcional que une HTML + atributos + JavaScript. O usuário digita o nome e clica no botão — a página responde sem recarregar!

saudacao.html — copie, cole no VS Code e abra no navegador!

```
<!DOCTYPE html>

<html>

<head>

<title>App de Saudacao</title>

</head>

<body>

<h1 id="titulo">Qual e o seu nome?</h1>

<input id="campo-nome" type="text" placeholder="Digite seu nome">

<button id="btn-saudar">Me cumprimente!</button>

<script>

// 1. Encontra os elementos pelo id

let btn = document.getElementById('btn-saudar');

let campo = document.getElementById('campo-nome');

let titulo = document.getElementById('titulo');

// 2. Escuta o clique no botao

btn.addEventListener('click', function() {

// 3. Le o que foi digitado

let nome = campo.value;

// 4. Modifica o titulo com o nome

titulo.textContent = 'Ola, ' + nome + '! Bem-vindo!';

titulo.style.color = 'green';

});

</script>

</body>

</html>
```

Como funciona passo a passo:

Passo	O que acontece
1. HTML carrega	O navegador monta a página com o título, o campo e o botão
2. JS carrega	O JS encontra os três elementos pelos IDs e fica 'de plantão'
3. Usuário digita	O usuário escreve o nome no campo <input>
4. Usuário clica	O evento 'click' dispara a função
5. JS lê o input	campo.value captura o texto digitado
6. JS modifica	titulo.textContent e titulo.style.color são atualizados na hora

7. Resumão

HTML	Atributos	JavaScript
<ul style="list-style-type: none">- Estrutura da pagina- Formado por tags- Tags abrem e fecham- Define o que existe	<ul style="list-style-type: none">- Ficam dentro da tag- Formato: nome="valor"- Exemplos: id, src, href, class, type	<ul style="list-style-type: none">- Faz a página reagir- Encontra elementos- Muda texto e estilo- Responde a cliques

Próximos passos sugeridos:

1. Crie o arquivo saudacao.html e abra no navegador
2. Faça um botão que muda a cor de fundo da pagina inteira ao clicar
3. Crie um campo que mostra o texto digitado em tempo real (evento 'input')
4. Pressione F12 no navegador para abrir o DevTools e inspecionar o HTML
5. Experimente mudar os valores no Console do DevTools em tempo real

Lembre-se: todo programador começa errando. O erro faz parte — use o Console do navegador (F12 > Console) para ler as mensagens de erro e entender o que deu errado. Boa prática!