

HTML

O que vamos ver hoje?

- Um pouco de *história*
- Estrutura HTML
- Principais tags

Um pouco de história

Surgimento

- A sua evolução começou em 1989, quando Tim Berners-Lee, um cientista da computação britânico que trabalhava no **CERN** (Organização Europeia para a Pesquisa Nuclear), desenvolveu a ideia da **World Wide Web**.
- A primeira versão do HTML, conhecida como HTML 1.0, foi lançada em 1991. Essa versão incluía tags básicas para formatar o texto, como **<h1>** para títulos, **<p>** para parágrafos e **<a>** para links. Também incluía imagens, embora apenas em preto e branco.

Surgimento

- Em 1997, o HTML 3.0 foi lançado, trazendo suporte para folhas de estilo (**CSS**) e elementos multimídia, como vídeo e som.
- Em 1999, foi lançado o HTML 4.0, que incluía novos recursos como tabelas avançadas, frames e suporte para scripts do lado do cliente (**JavaScript**).
- O **HTML5** foi lançado em 2014 e incluiu muitos novos recursos, como elementos semânticos (como <article> e <section>), suporte para vídeo e áudio, geolocalização e capacidade de armazenamento local.

O que é HTML?

- HTML (*Hypertext Markup Language* ou *Linguagem de Marcação de HiperTexto*) é uma **linguagem de marcação** (não de programação) da web.
- Pense no HTML como o **esqueleto de uma página**, ele é responsável pelos textos, links, listas e imagens.

Iniciando

Como começar?

- HTML é escrito em arquivos com o formato **.html**.
- Para criar uma página HTML é fácil, entre em seu editor de código (recomendo o Visual Studio Code) e salve o arquivo em branco como **index.html**.

Estrutura inicial

- A estrutura inicial do seu HTML será essa:



```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Estrutura inicial

- `<!doctype html>` - Essa tag (não tem fechamento) informa ao navegador que o arquivo faz parte de um documento HTML5.
- `<html>` - Representa a raiz de um documento HTML.

Estrutura inicial

- `<head>` - Contém informações sobre sua página, mas não o conteúdo que vai aparecer na tela da sua página. Ela conterá coisas como: links para folhas de estilo (CSS), título da página, links de fontes e meta tags e entre outros.
- `<body>` - Contém todo o conteúdo que vai aparecer na tela da sua página. Todo o código que você queira apresentar na página deverá estar dentro dele.

Meta tags

SEO

- SEO (*Search Engine Optimization*) é o conjunto de técnicas responsáveis por fazer com que determinada página de um site apareça nas primeiras páginas da ferramenta de busca.
- Quanto melhor for o SEO de um site, maiores as chances de que a página apareça logo de cara para quem pesquisa por determinado assunto.

Meta


- A tag **<meta>** define metadados (informações) sobre um documento HTML.
- As tags `<meta>` sempre vão dentro do elemento **<head>**.
- Normalmente são usadas para especificar o conjunto de caracteres, descrição da página, palavras-chave, autor do documento, configurações da janela de visualização.

Meta

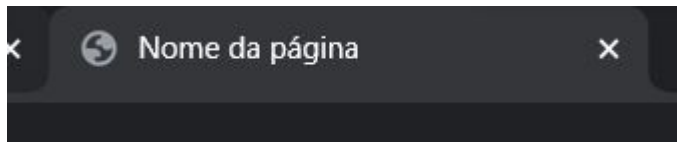
- Os metadados não serão exibidos na página, mas podem ser analisados por máquina.
- Os metadados são usados por navegadores, mecanismos de busca e outros serviços da web.

Title

- Define o título do documento. Ele é mostrado na aba da página do navegador.

A dark-themed code editor window with three colored window control buttons (red, yellow, green) in the top-left corner. It contains a single line of HTML code: `1 <title>Nome da página</title>`.

```
1 <title>Nome da página</title>
```




Description

- Contém a descrição do conteúdo da página, vários navegadores usam esta meta como descrição padrão da página quando é marcada (favoritos).

```
1 <meta name="description"  
2   content="Descrição sobre a página"  
3 />
```

Charset


- Define a codificação de caracteres usados na página. O conjunto de caracteres UTF-8 cobre quase todos os caracteres e símbolos do mundo.
- Sempre tenha ele em suas páginas.



```
1 <meta charset="UTF-8">
```

Keywords

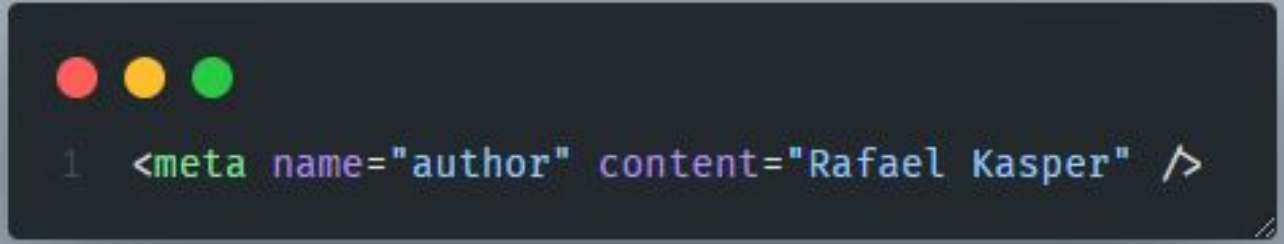
- Palavras-chaves associadas ao conteúdo da página, contendo strings separadas por vírgulas.



```
1 <meta name="keywords" content="css, html, js">
```

Author

- Define o autor do documento/site.



```
1 <meta name="author" content="Rafael Kasper" />
```

Tags principais

Sintaxe

- Os elementos HTML são escritos usando tags. Todas as tags tem uma chave de abertura e fechamento. A tag de fechamento tem uma barra após o primeiro colchete.



```
1  <p>Programador sem café é um poeta sem poesia</p>
```

Títulos

- As tags de títulos são representadas em até seis níveis. **<h1>** é o nível mais alto e **<h6>** é o mais baixo.
- Quanto maior for o nível da tag, maior vai ser o tamanho da fonte apresentada na tela e sua importância no SEO.

Títulos



```
1  <h1>Título do h1</h1>
2  <h2>Título do h2</h2>
3  <h3>Título do h3</h3>
4  <h4>Título do h4</h4>
5  <h5>Título do h5</h5>
6  <h6>Título do h6</h6>
```

Título do h1

Título do h2

Título do h3

Título do h4

Título do h5

Título do h6

Textos

- As tags de texto, definem diferentes formatações para diversos tipos de texto.
- Desde estilos de fonte, parágrafos, quebra de linha ou até mesmo spans. Iremos conhecê-las:

Textos

- `<p>` - Sendo a principal tag de texto, é usada para criar um **parágrafo**.
- `` - Geralmente são utilizados para **agrupar** uma pequena informação.
- `` - Deixa o texto em **negrito**.
- `<i>` - Deixa o texto em **itálico**.
- `<hr>` - Cria uma **linha** horizontal.
- `` - Adiciona **importância** (SEO) no texto.

Textos

Este é um parágrafo

Este é um span

Este é um parágrafo com o texto em **negrito**

Este é um parágrafo com o texto em *itálico*

Este é um parágrafo **com destaque para SEO**



```
1 <hr>
2 <p>Este é um parágrafo</p>
3 <span>Este é um span</span>
4 <p>Este é um parágrafo com o texto em <b>negrito</b></p>
5 <p>Este é um parágrafo com o texto em <i>itálico</i></p>
6 <p>Este é um parágrafo <strong>com destaque para SEO</strong></p>
7 <hr>
```

Imagem

- Para adicionar uma imagem na página é bem simples:



```
1  
```

Imagem



```
1 
```



Imagem



```
1 
```



Fixação

- Html é uma linguagem de marcação de texto que usa tags (etiquetas) para definir o que será exibido.
- Para textos temos as tags `<h1>` a `<h6>`, `<p>`, ``, ``, `<i>`, `` e `<hr>`.
- Para exibição de imagens usamos a tag ``

Organização de elementos em tela

Div

- É utilizada para **dividir elementos para fins de estilos** (usando class ou id). Ele deve ser utilizado somente quando não tiver outro elemento de semântica.
- É usada para uma divisão de um conteúdo pois, como o uso dela ajuda a **quebrar os elementos em linhas**, deixando melhor a visualização.

Div



```
1 <div>
2   <p>Entrou dentro de uma div</p>
3 </div>
```

← → ↻ ⓘ 127.0.0.1:5500/index.html

Entrou dentro de uma div

Link

- A tag **<a>** (ou elemento âncora) é um hiperlink, que é usado para **vincular uma página a outra**.
- O conteúdo dentro de cada **<a>** precisará indicar o destino do link utilizando o atributo href.

Link



```
1 <a href="https://www.senacrs.com.br/">Visitar Senac RS</a>
```

← → ↻ ⓘ 127.0.0.1:5500/index.html

[Visitar Senac RS](https://www.senacrs.com.br/)

Tabela

- A criação de uma tabela em uma página HTML é feita com a tag **<table>**.
- Em todas as tabelas existem linhas. Para isso, utilizamos a tag **<tr>**.
- A tag é utilizada para inserir o cabeçalho na tabela.
- Além das linhas, toda tabela precisa de uma coluna. É preciso adicionar a tag **<td>** em todas as linhas para criar as colunas desejadas e inserir o conteúdo a ser exibido.

Tabela



```
1  <table>
2    <tr>
3      <th>Nome</th>
4      <th>Sobrenome</th>
5      <th>Idade</th>
6    </tr>
7    <tr>
8      <td>Rafael</td>
9      <td>Kasper</td>
10     <td>37</td>
11   </tr>
12 </table>
```

Nome	Sobrenome	Idade
Rafael	Kasper	37

Listas

- O elemento **** é usado para representar um item de uma lista. Deve estar contido em um elemento pai: uma lista ordenada (****) ou uma lista desordenada (****).
- As listas **ordenadas** (ou numeradas) são usadas para indicar alguma **sequência ou numeração**.
- As listas **desordenadas** são usadas para **listar** itens, sem se preocupar com sua sequência.

Lista ordenada



```
1  <ol>
2    <li>HTML</li>
3    <li>CSS</li>
4    <li>Javascript</li>
5    <li>React</li>
6    <li>API</li>
7  </ol>
```

1. HTML
2. CSS
3. Javascript
4. React
5. API

Lista não-ordenada



```
1 <ul>
2   <li>HTML</li>
3   <li>CSS</li>
4   <li>Javascript</li>
5   <li>React</li>
6   <li>API</li>
7 </ul>
```

- HTML
- CSS
- Javascript
- React
- API

Fixação

- Div é uma forma de separar os conteúdos.
- Para listas ordenadas utilizamos a tag `` e para não ordenadas a tag ``
- Podemos construir tabelas de diferentes formas e tamanhos.

Entrada de dados

Input

- É usado para criar controles interativos que receber dados do usuário.
- A semântica de um **<input>** varia consideravelmente dependendo do valor de seu atributo **type**.
- O tipo padrão de um input é **text**, se este atributo não for especificado.

Input

- Types possíveis:
 - button, checkbox, color, date, email, file, password, search, submit.

Input

The diagram illustrates the mapping between HTML `<input type="..."/>` tags and their visual representation in a web browser. On the left, a code editor window displays nine lines of HTML code. On the right, the rendered UI elements are shown. Teal arrows connect each line of code to its respective UI component.

HTML Code	UI Element
<code><input /></code>	Text input field (containing "Texto")
<code><input type="button" /></code>	Small square button
<code><input type="color" /></code>	Color selection field (black square)
<code><input type="date" /></code>	Date input field (containing "dd/mm/aaaa" and a calendar icon)
<code><input type="email" /></code>	Email input field (containing "email")
<code><input type="file" /></code>	File selection interface (button "Escolher arquivo" and text "Nenhum arquivo escolhido")
<code><input type="password" /></code>	Password input field (containing seven dots)
<code><input type="search" /></code>	Search input field (containing "busca")
<code><input type="submit" /></code>	Submit button (containing "Enviar")

Label

- É a maneira formal de definir um rótulo.
- Geralmente utilizado em conjunto com `<input>`



```
1 <label>Nome: <input /></label>
```

Nome:

Formulários

- Um formulário HTML é feito de um ou mais widgets.
- Esses widgets podem ser campos de texto, caixas de seleção, botões, checkboxes, radio buttons e entre outros elementos HTML.

Formulários

- O elemento **<form>** define um formulário e os seus atributos definem a maneira como esse formulário se comporta.



```
1 <form action="/" method="post"></form>
```

Formulários

- O elemento **<fieldset>** é usado para agrupar elementos em um formulário.



```
1 <fieldset></fieldset>
```

Formulários

- Exemplo de formulário:



```
1 <form action="/" method="post">
2   <label>Nome:</label>
3   <input type="text" />
4   <label>Email:</label>
5   <input type="email" />
6   <input type="submit" value="Enviar" />
7 </form>
```

Nome: Email:

Fixação

- Input serve para a entrada de dados, definidos por seu type.
- Os tipos possíveis são button, checkbox, color, date, email, file, password, search, submit.
- Label é uma forma de inserir um rótulo ao input.

Tags semânticas

Conceito

- As tags semânticas além de deixar o código melhor para o **SEO**, ele ajuda outros desenvolvedores a **entender seu código**.
- Não têm nenhum efeito na apresentação na página.
- Elementos semânticos: <nav>, <aside>, <article>, <footer>, <section>, entre outros.
- Elementos não semânticos: <div> e .

Section

- É utilizada para **marcar as seções** de conteúdo de uma página.
- Com esse elemento agrupamos de forma lógica nosso conteúdo, separando a informação em áreas diferentes.
- O principal objetivo é retirar essa responsabilidade das divs.

```
1 <section></section>
```


Article

- Representa uma composição **independente em um documento**. Isso é, não precisa do resto do site para contextualizar.
- Este poderia ser um simples card, um artigo de revista ou jornal, um post de um blog, ou qualquer outra forma de conteúdo independente.

```
1 <article></article>
```

Aside

- Representa uma **seção** de uma página que consiste de conteúdo que poderia ser considerado **separado do conteúdo principal**.
- Essas seções são, muitas vezes, representadas como barras laterais.

```
1 <aside></aside>
```

Header

- Representa um container para **conteúdo introdutório** (muitas vezes usado na primeira seção da página).



```
1 <header></header>
```

Footer

- Representa um **rodapé** de um conteúdo.



```
1 <footer></footer>
```

Nav

- Representa uma **seção** de uma página que **aponta para outras páginas** ou para **outras áreas** da página, ou seja, uma seção com grupos de links de navegação.



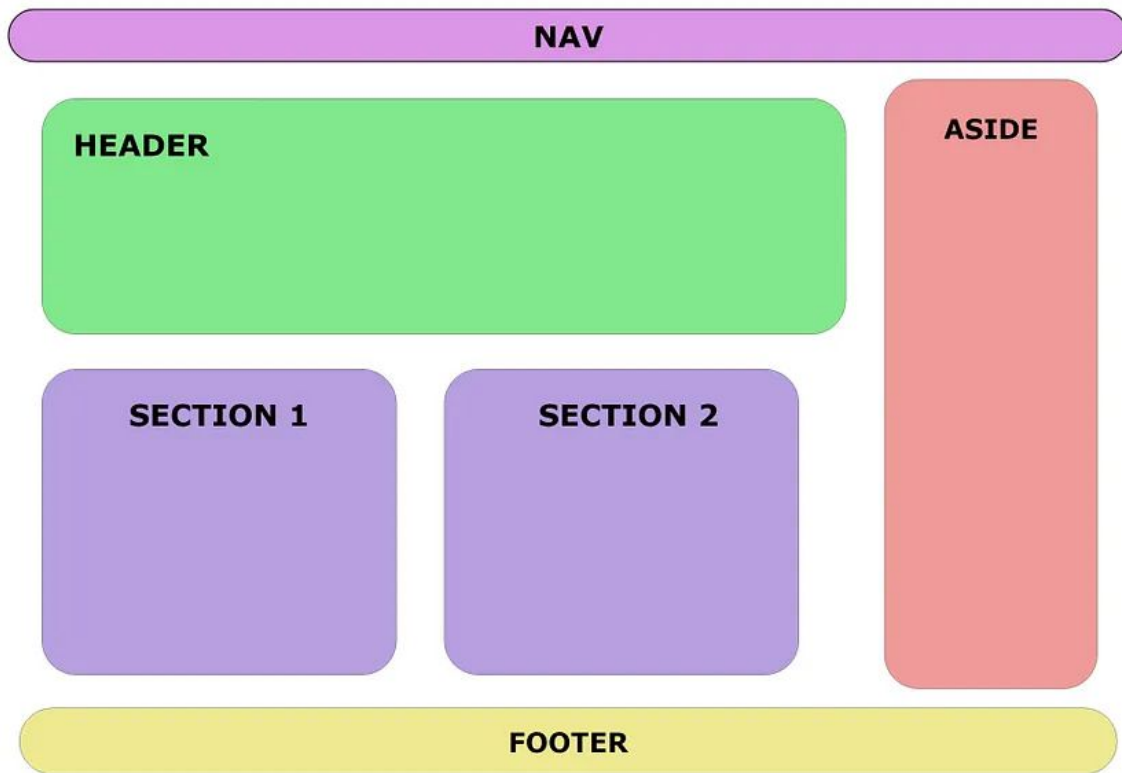
```
1 <nav></nav>
```

Main

- Define o **conteúdo principal** da página.
- Ele representa o conteúdo mais importante da página, que está diretamente relacionado ao tópico central do documento.
- É importante **utilizar apenas uma vez** esse elemento por arquivos HTML criado.

```
1 <main></main>
```

Estrutura padrão



Estrutura padrão

```
1 <header>
2   <h1>Cursos de programação</h1>
3 </header>
4 <main>
5   <section>
6     <h2>React</h2>
7     <article>
8       <h3>Curso de Componentes em React</h3>
9       <p>
10        Neste curso você aprenderá a desenvolver aplicações componentizadas
11        com React.
12      </p>
13    </article>
14    <article>
15      <h3>Curso de Typescript</h3>
16      <p>
17        Neste curso você aprenderá utilizar o superset Typescript em
18        aplicações React.
19      </p>
20    </article>
21  </section>
22  <section>
23    <h2>Javascript</h2>
24    <article>
25      <h3>Curso básico de Javascript</h3>
26      <p>Conheça a linguagem de programação Javascript.</p>
27    </article>
28    <article>
29      <h3>Deixando suas páginas dinâmicas</h3>
30      <p>Aprenda a criar páginas dinâmicas com Javascript.</p>
31    </article>
32  </section>
33 </main>
34 <footer>
35   <p>MinhaEmpresa.com.br. Todos os direitos reservados.</p>
36 </footer>
```

Cursos de programação

React

Curso de Componentes em React

Neste curso você aprenderá a desenvolver aplicações componentizadas com React.

Curso de Typescript

Neste curso você aprenderá utilizar o superset Typescript em aplicações React.

Javascript

Curso básico de Javascript

Conheça a linguagem de programação Javascript.

Deixando suas páginas dinâmicas

Aprenda a criar páginas dinâmicas com Javascript.

MinhaEmpresa.com.br. Todos os direitos reservados.

Fixação

- Tags semânticas ajudam as ferramentas de SEO e ao entendimento do código por outros programadores.

Programa

3000 TALENTOS TI

Vamos Pesquisar!

Conheça mais tags

- Acesse os links abaixo e conheça mais tags HTML:

<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>

https://www.w3schools.com/html/html_intro.asp

Resumo

Resumo



- HTML é a linguagem de marcação usada para criar páginas da web. Ele consiste em tags (etiquetas) que definem a estrutura e o conteúdo de uma página.
- A tag `<html>` define o início e o fim do documento, enquanto `<head>` contém informações como o título e metadados. O título da página é especificado com `<title>`.

Resumo

- O conteúdo visível é colocado dentro da tag `<body>`, onde podemos estruturá-lo com cabeçalhos `<h1>`, `<h2>`, etc., e parágrafos usando `<p>`.
- Links são criados com `<a>`, imagens com ``, e listas com ``, ``, e ``.
- Tags como `<div>` e `` são usadas para agrupar e estilizar elementos.
- Tabelas com `<table>`, `<tr>`, `<th>` e `<td>`.

Dúvidas?



Programa
3000 TALENTOS TI
Obrigado(a)!