

Introdução ao HTML e JavaScript

<LAB365>

SENAI

AGENDA

- O que é HTML
- A estrutura de uma página HTML
- Elementos comuns
- Introdução ao JavaScript

O QUE É HTML

HTML (*Hypertext Markup Language*, ou **linguagem de marcação de hipertexto**) é definida como uma linguagem de marcação utilizada para estruturar uma página da web.

Linguagens de marcação **não** são linguagens de **programação**.

O QUE É HTML

O HTML é padronizado pela **World Wide Web Consortium**, um consórcio internacional de mais de 450 empresas e órgãos que criam padrões para diversas ferramentas utilizadas no desenvolvimento web.

Link para o site da organização: <https://www.w3.org/>

O QUE É HTML

O HTML possui uma coleção de elementos que utilizamos para criar nossas páginas.

Uma página HTML padrão possui um elemento que identifica a **versão do HTML** a ser utilizada (hoje, HTML5), os **metadados da página**, o **título** e o **corpo da página**.

O QUE É HTML

Exemplo:

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta name="description" content="Descrição da página">
6      <title>Título da minha página</title>
7  </head>
8
9  <body>
10     Corpo da página
11 </body>
12
13 </html>
14
```

O QUE É HTML

O **VSCode** possui uma ferramenta integrada chamada **emmet**, que nos auxilia a escrever nosso código, criando desde elementos simples até blocos completos.

Por exemplo, para criarmos um código padrão de HTML, basta digitarmos `"html:5"` (ou simplesmente `"!"`) e, quando o diálogo do emmet aparecer, apertamos **enter** ou **tab**.

Mãos à obra!



ESTRUTURA DE UMA PÁGINA

A estrutura mínima de um documento HTML é formada pelas **tags**:

- Doctype - versão do HTML utilizada
- html - abertura do documento html e especificação da língua
- head - metadados do documento. Não aparece na página
- body - corpo do documento. Aparece na página.

ESTRUTURA DE UMA PÁGINA

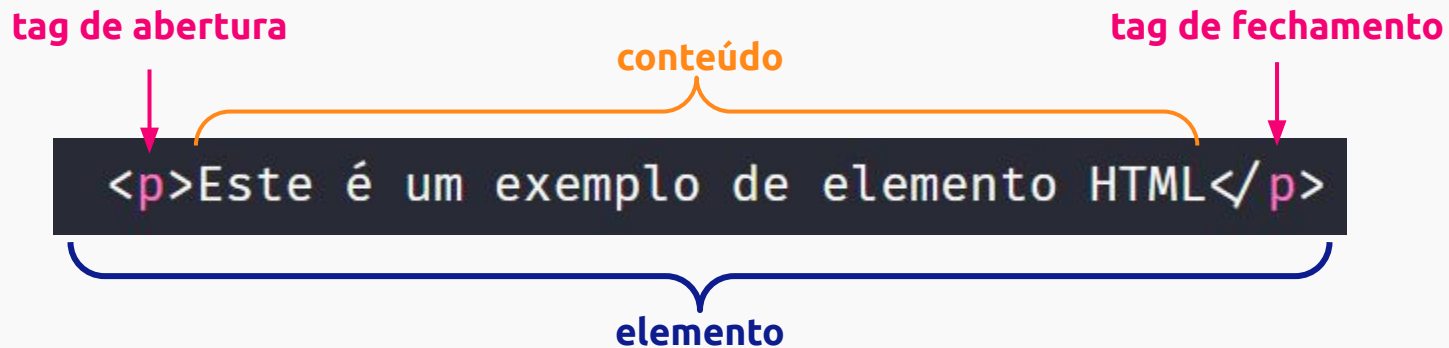
Um típico **elemento HTML** é formado por **tags**. Cada tag HTML possui um conjunto de **atributos** (*attributes*), sendo que algumas tags possuem atributos **obrigatórios**.

Uma tag sempre precisa ser **fechada** (apesar de o navegador ser tolerante quanto a isso).

Podemos ter **elementos vazios** ou **elementos com conteúdo**.

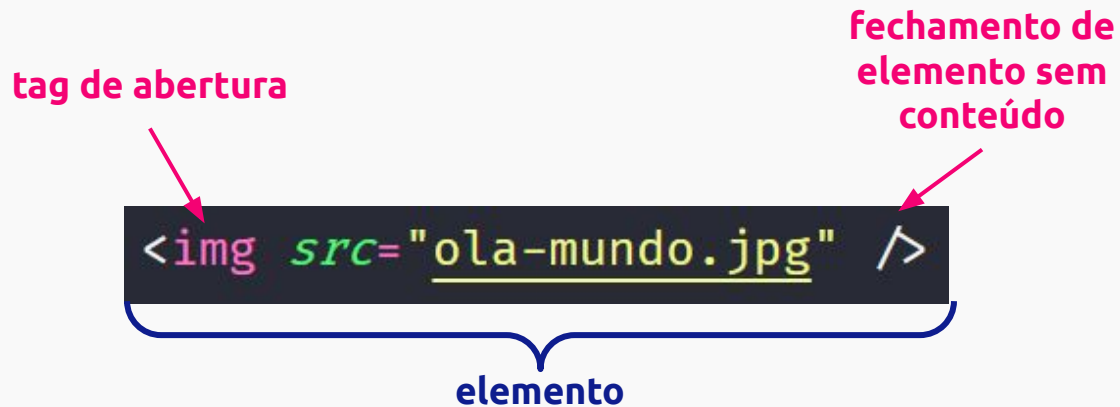
ESTRUTURA DE UMA PÁGINA

Exemplos:



ESTRUTURA DE UMA PÁGINA

Exemplos:



ESTRUTURA DE UMA PÁGINA

Elementos HTML podem ser **aninhados** (inseridos um dentro do outro). De fato, é muito comum aninhar elementos para estruturar nossas páginas.

Devemos, porém, sempre observar a ordem de **abertura e fechamento** das tags nos elementos aninhados.

ESTRUTURA DE UMA PÁGINA

Exemplos:

tag de abertura
do <p>

tag de abertura
do

tag de fechamento do
<p>

```
<p>Este é um exemplo de elemento <b>HTML</b></p>
```

tag de fechamento
do

- **Atributos**

Os **atributos** adicionam informações às **tags**, que podem ajudar a identificar um elemento específico ou um conjunto de elementos, adicionar semântica ao elemento ou até alterar sua visualização.

Atributos são **sempre** aplicados à tag de **abertura**.

ESTRUTURA DE UMA PÁGINA

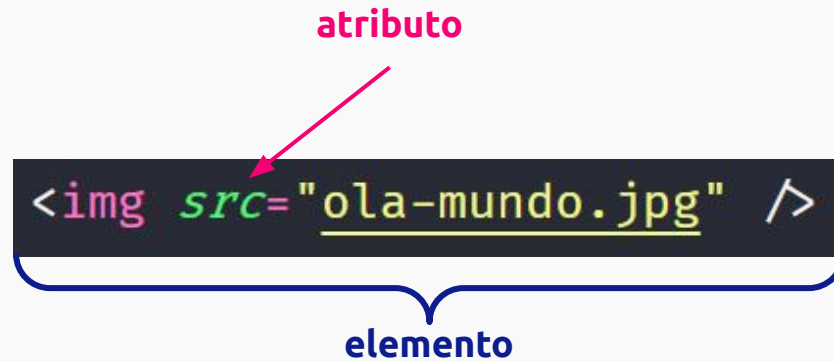
Exemplo:

atributo

```

```

elemento

The diagram shows an HTML element `` on a dark background. A pink arrow points from the word "atributo" to the `src` attribute. A blue bracket underneath the entire code snippet points to the word "elemento".

Mãos à obra!



Existem muitas tags HTML. Para facilitar nossa busca pela tag, podemos pensar em categorias:

- Texto (*h1 - h6, p, strong, span, etc*)
- Hiperlink (*a*)
- Multimídia (*img, video, audio, etc*)
- Formulário (*form, fieldset, input, button, etc*)
- Tabela (*table, thead, tbody, th, td, tr, etc*)
- Genérico (*div, span*)

ELEMENTOS COMUNS

- Texto

```
<h1>Titulo principal</h1>
<p>Lorem ipsum dolor sit amet.</p>

<h2>Título secundário</h2>
<p>
  Lorem ipsum, dolor sit amet consectetur adipisicing elit.
  Porro, <strong>sapiente</strong>.
</p>
```

ELEMENTOS COMUNS

- Hiperlink

```
<a href="http://br.linkedin.com">Página do linkedin</a>
```

```
<a href="#sobre-mim">Sobre mim</a>
```

ELEMENTOS COMUNS

- Multimídia

```

```

ELEMENTOS COMUNS

- Formulário

```
<form>
  <fieldset>
    <label for="nome">Nome:</label>
    <input type="text" name="nome" id="nome" />

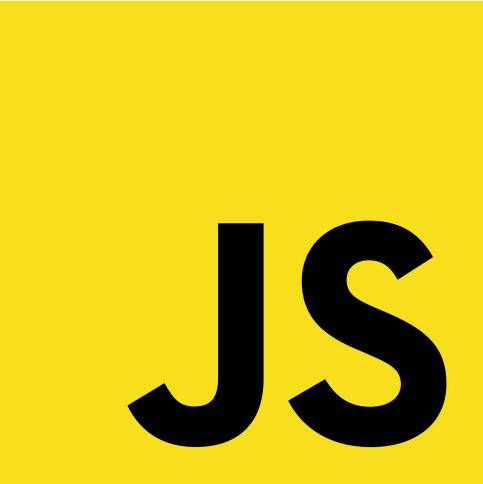
    <label for="sobrenome">Sobrenome:</label>
    <input type="text" name="sobrenome" id="sobrenome" />
  </fieldset>
</form>
```

- Tabela

```
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Nome</th>
      <th>Filme preferido</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>1</th>
      <th>Michael</th>
      <th>Armagedon</th>
    </tr>
  </tbody>
</table>
```

Mãos à obra!



A yellow square with the letters 'JS' in a bold, black, sans-serif font.

O **JavaScript** é uma **linguagem de programação** criada em meados de **1995** por Brendan Eich. Seu nome, inicialmente, era *Mocha*. Foi criada para trazer interatividade aos recém criados navegadores web.

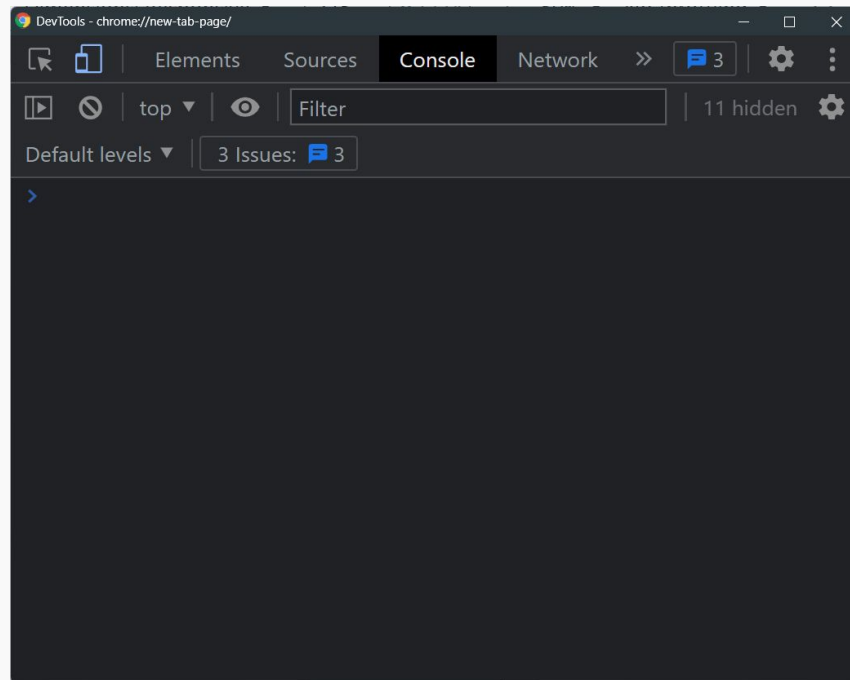
*Curiosidade: todo o protótipo da primeira versão do **JS** ficou pronto em **10 dias!***

INTRODUÇÃO AO JAVASCRIPT

Developer Tools

- F12 do teclado
- Botão direito + inspecionar
- CTRL + SHIFT + i

Verifique se a aba selecionada é a "Console"



INTRODUÇÃO AO JAVASCRIPT

Console.log é um **método** JavaScript utilizado para imprimir no console do navegador alguma informação.

A sua sintaxe é:

```
console.log ("Olá mundo!")
```

INTRODUÇÃO AO JAVASCRIPT

Usamos **variáveis** em nosso código sempre que queremos **armazenar um valor** para ser reutilizado posteriormente.

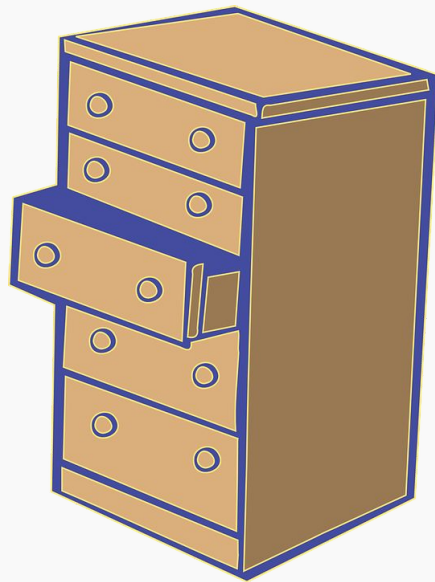
Uma variável funciona como uma “**etiqueta**” que indica o endereço do valor atribuído à ela na memória do nosso computador.

INTRODUÇÃO AO JAVASCRIPT

Podemos imaginar a memória do computador como um enorme “armário” cheio de “gavetas”.

Guardamos nosso valor “Olá mundo!” em uma gaveta de nome “**primeiraVariavel**”.

```
var primeiraVariavel = “Olá mundo!”
```



INTRODUÇÃO AO JAVASCRIPT

Toda vez que quisermos recuperar o valor dessa variável, basta passarmos o nome dela para a operação que estamos fazendo:

```
console.log( primeiraVariavel )
```

- **VAR:**
 - É uma variável que podemos atribuir um novo valor para ela (reatribuição de valor)
 - É possível criar outra variável com o mesmo nome (redeclaração de variável)

- **LET:**
 - É uma variável que podemos atribuir um novo valor para ela (reatribuição de valor)
 - Não podemos criar mais de uma variável com o mesmo nome (redeclaração de variável)

- **CONST:**

- É uma variável que não pode ter o seu valor inicial alterado (reatribuição de valor)
- Não podemos criar mais de uma variável com o mesmo nome (redeclaração de variável)
- É obrigatório inicializar o valor de uma **const** em sua declaração

INTRODUÇÃO AO JAVASCRIPT

Existem algumas regras para determinarmos o nome de uma variável:

- Só pode conter **letras, dígitos** ou os símbolos \$ e _;
- O primeiro caractere **não** pode ser um **dígito**;
- Não podemos utilizar palavras reservadas da linguagem;

Podemos utilizar duas convenções para criar nomes de variáveis compostos: o padrão **snake_case** ou o **camelCase**.

INTRODUÇÃO AO JAVASCRIPT

Nossas variáveis podem armazenar diversos tipos de dados que são definidos pelo JavaScript.

Uma variável no JavaScript pode ser sobrescrita com tipos diferentes de dados - característica que costumamos chamar de **linguagem de tipagem fraca**.

Entender os tipos de dados é importante para a boa implementação do código e para evitar erros.

INTRODUÇÃO AO JAVASCRIPT

Estes são os tipos de dados principais:

- **String**
- **Number**
- **Boolean**
- **Null**
- **Undefined**

INTRODUÇÃO AO JAVASCRIPT

Mãos à obra!





OBRIGADO!

<LAB365>