

# HTML

## Sumário

|                                    |          |
|------------------------------------|----------|
| <b>TAG .....</b>                   | <b>1</b> |
| <b>ESTRUTURA BASICA.....</b>       | <b>1</b> |
| <!DOCTYPE HTML> .....              | 1        |
| <HTML> .....                       | 1        |
| <HEAD> .....                       | 1        |
| <TITLE> .....                      | 1        |
| <META> .....                       | 1        |
| <BODY> .....                       | 2        |
| <SCRIPT>.....                      | 2        |
| <LINK> .....                       | 2        |
| <STYLE> .....                      | 2        |
| <b>TÍTULOS E PARÁGRAFOS .....</b>  | <b>2</b> |
| <H1>, <H2><H6>.....                | 2        |
| <P> .....                          | 2        |
| <b>COMENTÁRIOS.....</b>            | <b>3</b> |
| <b>FORMATAÇÃO.....</b>             | <b>3</b> |
| <B> E <STRONG> .....               | 3        |
| <I> E <EM> .....                   | 3        |
| <MARK> .....                       | 4        |
| <SMALL>.....                       | 4        |
| <DEL>.....                         | 4        |
| <INS> .....                        | 4        |
| <SUB> .....                        | 4        |
| <SUP> .....                        | 4        |
| <HR> .....                         | 4        |
| <b>ATRIBUTOS.....</b>              | <b>5</b> |
| <b>LINKS .....</b>                 | <b>5</b> |
| ATRIBUTOS DE DESTINO .....         | 6        |
| TÍTULOS DE LINKS.....              | 6        |
| <b>IMAGENS .....</b>               | <b>7</b> |
| MAPAS DE IMAGEM .....              | 7        |
| <b>QUEBRA DE LINHA .....</b>       | <b>8</b> |
| <b>LISTAS .....</b>                | <b>8</b> |
| LISTA HTML NÃO ORDENADA <UL> ..... | 8        |
| LISTA HTML ORDENADA <OL> .....     | 8        |

|                                       |           |
|---------------------------------------|-----------|
| LISTAS DE DESCRIÇÃO <DL>.....         | 9         |
| <b>TABELAS.....</b>                   | <b>9</b>  |
| DEFINIR UMA TABELA HTML <TABLE> ..... | 9         |
| LINHAS DA TABELA <TR> .....           | 9         |
| CÉLULAS DA TABELA <TD>.....           | 9         |
| CABEÇALHOS DE TABELA <TH>.....        | 10        |
| <b>ELEMENTO &lt;DIV&gt; .....</b>     | <b>10</b> |
| <DIV> E <SPAN>.....                   | 10        |
| <b>FORMULÁRIOS HTML.....</b>          | <b>11</b> |
| ATRIBUTO ACTION.....                  | 11        |
| ATRIBUTO METHOD .....                 | 11        |
| ATRIBUTO TARGET .....                 | 12        |
| ATRIBUTO AUTOCOMPLETE.....            | 13        |
| ATRIBUTO NOVALIDATE .....             | 13        |
| <b>ELEMENTOS DE FORMULÁRIO.....</b>   | <b>13</b> |
| ELEMENTO <INPUT> .....                | 13        |
| ELEMENTO <LABEL> .....                | 13        |
| ELEMENTO <SELECT>.....                | 14        |
| ELEMENTO <TEXTAREA>.....              | 14        |
| ELEMENTO <BUTTON> .....               | 14        |
| ELEMENTOS <FIELDSET> E <LEGEND> ..... | 15        |
| ELEMENTO <DATA LIST> .....            | 15        |
| ELEMENTO <OUTPUT> .....               | 15        |
| <b>TIPOS DE INPUT .....</b>           | <b>16</b> |
| <INPUT TYPE="TEXT"> .....             | 16        |
| <INPUT TYPE="PASSWORD">.....          | 16        |
| <INPUT TYPE="SUBMIT">.....            | 16        |
| <INPUT TYPE="RESET"> .....            | 16        |
| <INPUT TYPE="RADIO"> .....            | 17        |
| <INPUT TYPE="CHECKBOX "> .....        | 17        |
| <INPUT TYPE="BUTTON">.....            | 17        |
| <INPUT TYPE="COLOR"> .....            | 17        |
| <INPUT TYPE="DATE "> .....            | 17        |
| <INPUT TYPE="DATETIME-LOCAL"> .....   | 18        |
| <INPUT TYPE="MONTH"> .....            | 18        |
| <INPUT TYPE="WEEK"> .....             | 18        |
| <INPUT TYPE="TIME">.....              | 18        |
| <INPUT TYPE="EMAIL "> .....           | 18        |
| <INPUT TYPE="FILE ">.....             | 18        |

|  |           |
|--|-----------|
| <INPUT TYPE="HIDDEN ">.....                | 18        |
| <INPUT TYPE="NUMBER"> .....                | 19        |
| <INPUT TYPE="RANGE"> .....                 | 19        |
| <INPUT TYPE="SEARCH"> .....                | 19        |
| <INPUT TYPE="TEL"> .....                   | 19        |
| <INPUT TYPE="URL">.....                    | 19        |
| <b>ELEMENTOS SEMÂNTICOS .....</b>          | <b>20</b> |
| ELEMENTO <SECTION>.....                    | 20        |
| ELEMENTO <ARTICLE> .....                   | 20        |
| ELEMENTO <HEADER >.....                    | 20        |
| ELEMENTO <FOOTER> .....                    | 20        |
| ELEMENTO <NAV>.....                        | 21        |
| ELEMENTO <ASIDE>.....                      | 21        |
| ELEMENTOS <FIGURE> E <FIGCAPTION>.....     | 21        |
| <b>HTML ATRIBUTOS PERSONALIZADAS .....</b> | <b>21</b> |
| <b>HTML MULTIMÍDIA .....</b>               | <b>23</b> |
| HTML VIDEO.....                            | 23        |
| HTML AUDIO .....                           | 24        |
| PROPRIEDADES AUDIO/VÍDEO .....             | 24        |
| EVENTOS AUDIO/VÍDEO.....                   | 25        |
| MÉTODOS AUDIO/VÍDEO.....                   | 26        |
| <b>IFRAME .....</b>                        | <b>26</b> |
| ATRIBUTOS IFRAME .....                     | 26        |
| <b>HTML CANVAS.....</b>                    | <b>28</b> |
| DESENHANDO NO CANVAS COM JS.....           | 28        |
| COORDENADAS CANVAS .....                   | 29        |
| RETÂNGULO CANVAS .....                     | 29        |
| PATH CANVAS .....                          | 30        |
| CÍRCULO CANVAS.....                        | 31        |
| IMAGENS CANVAS .....                       | 32        |
| ANIMAÇÕES CANVAS .....                     | 32        |

## TAG

As tags são formadas por uma estrutura própria, iniciam com o sinal “maior que”, em seguida vem o nome daquele elemento e por fim, o sinal “menor que”. Podem ser dispostas em tags que precisam de fechamento e tags que fecham sozinhas (self-closing). O fechamento de uma tag será definido com uma barra (/), sendo que no caso das tags de auto fechamento, não há necessidade da presença desse caractere.

## ESTRUTURA BASICA

### **<!DOCTYPE html>**

Informa ao navegador que esse documento é do tipo HTML e sua versão.

### **<html>**

Representa a raiz do documento, serve com um container que engloba todos os outros elementos HTML.

### **<head>**

Compreende as informações do documento que serão interpretadas pelo navegador (metadados). Como por exemplo, título do documento, links para folhas de estilo etc.

### **<title>**

Define o título do documento. O título deve ser somente texto e é mostrado na barra de título do navegador ou na aba da página. A tag <title> é obrigatória em documentos HTML. O conteúdo de um título de página é muito importante para a otimização de mecanismos de busca (SEO)! O título da página é usado pelos algoritmos do mecanismo de pesquisa para decidir a ordem ao listar as páginas nos resultados da pesquisa.

### **<meta>**

Define metadados, ou seja, informações sobre dados de um documento HTML. As <meta> tags vão dentro do elemento <head> e são usadas para especificar o conjunto de caracteres, o autor do documento, as configurações da

janela de visualização etc. A tag <meta> define metadados sobre um documento HTML. Metadados são dados (informações) sobre dados.

### **<body>**

É onde fica todo o conteúdo de texto, imagem e vídeos, em que o usuário vê e interage, nele os conteúdos são estruturados pelas demais tags do HTML.

### **<script>**

Esse elemento contém instruções de script ou aponta para um arquivo de script externo por meio do atributo src.

### **<link>**

É uma tag vazia, que contém apenas atributos e faz a relação do documento HTML com recursos externos, é comumente usado para vincular uma folha de estilo externa, também é usada para definir o favicon da página (ícone da aba do navegador), como outros recursos.

### **<style>**

Essa tag é usada para declarar estilos (CSS) para um documento.

## **TÍTULOS E PARÁGRAFOS**

### **<h1>, <h2><h6>**

A família de cabeçalhos ou headings, define os títulos dos conteúdos. Esse grupo de elementos possuem um alto valor semântico e uma organização hierárquica, indo de <h1> até <h6>, sendo o h1 de maior valor semântico e o h6 de menor valor semântico.

### **<p>**

A tag <p> define um parágrafo. Os navegadores adicionam automaticamente uma única linha em branco antes e depois de cada elemento <p>.

## COMENTÁRIOS

Comentários no HTML ou em qualquer outra linguagem são notas que podem ser incluídas no código fonte para descrever o que se quiser. Assim, não modificam o programa executado e servem somente para ajudar a pessoa que está desenvolvendo a melhor organizar os seus códigos. Para comentar algum código no HTML você deve envolvê-lo entre `<!-- Seu código aqui -->`. Você também pode ocultar mais de uma linha, tudo entre `<!--` e `-->` será ocultado da tela.

## FORMATAÇÃO

### `<b>` e `<strong>`

Ambos definem partes de texto como negrito (bold), porém se diferenciam em seu nível de importância.

O elemento HTML `<b>` define que um texto em negrito (bold), porém sem nenhuma importância.

O elemento HTML `<strong>` define o texto com forte (strong) importância. Essa diferenciação é muito importante para a otimização de mecanismos de busca. Ex: `<strong>texto importante</strong><b>texto em negrito</b>`.

### `<i>` e `<em>`

O elemento HTML `<i>` define uma parte do texto em uma voz ou humor alternativo. O conteúdo interno geralmente é exibido em itálico. A tag `<i>` é frequentemente usada para indicar um termo técnico, uma frase de outro idioma, um pensamento, um nome de navio, etc.

O elemento HTML `<em>` define o texto enfatizado. O conteúdo interno geralmente é exibido em itálico. O leitor pronunciará as palavras em `<em>` com ênfase, usando acento verbal.

Ou seja, um define o texto como itálico e o outro define o texto que deve ser lido com ênfase. Ex: `<i>italic text</i><em>texto com ênfase</em>`.

**<mark>**

O elemento HTML <mark> define o texto que deve ser marcado, realçado ou destacado. Esse elemento vai colocar uma cor de fundo no texto definido.

Ex: <mark>texto marcado</mark>.

**<small>**

O elemento HTML <small> define um texto menor, normalmente utilizado em situações específicas. Ex: <small>texto em small</small>.

**<del>**

Define o texto que foi excluído de um documento. Os navegadores geralmente riscam uma linha excluída de um texto. Ex: <del>texto excluído</del>.

**<ins>**

Elemento O elemento HTML <ins> define um texto que foi inserido em um documento. Os navegadores geralmente sublinham o texto inserido. Ex <ins>texto inserido</ins>.

**<sub>**

Define o texto subscrito. O texto subscrito aparece meio caractere abaixo da linha normal e às vezes é renderizado em uma fonte menor. O texto subscrito pode ser usado para fórmulas químicas, como H<sub>2</sub>O. Ex: <sub>texto subscrito</sub>.

**<sup>**

Define o texto sobrescrito. O texto sobrescrito aparece meio caractere acima da linha normal e, às vezes, é renderizado em uma fonte menor. O texto sobrescrito pode ser usado para notas de rodapé por exemplo como www<sup>[1]</sup>. Ex: <sup>texto sobrescrito</sup>.

**<hr>**

representa uma quebra temática entre elementos de nível de parágrafo (por exemplo, uma mudança da cena de uma história, ou uma mudança de tema com uma seção). Nas versões anteriores do HTML, representava uma linha horizontal. Pode continuar sendo exibida como uma linha horizontal nos

navegadores, mas agora está definida em termos semânticos, em vez de termos de apresentação.

Ex: `<p>Este é o primeiro parágrafo do texto.</p> <hr> <p> Este é o segundo parágrafo do texto.</p>`

Este é o primeiro parágrafo do texto.

---

Este é o segundo parágrafo do texto.

## ATRIBUTOS

Os atributos HTML fornecem informações adicionais sobre elementos HTML. Todos os elementos HTML podem ter atributos Os atributos fornecem informações adicionais sobre os elementos Os atributos são sempre especificados na tag inicial Os atributos geralmente vêm em pares com nome e valor como: `name="value"`.

## LINKS

Os links são encontrados em quase todas as páginas da web. Os links permitem que os usuários cliquem de uma página para outra. Links HTML são hiperlinks. Você pode clicar em um link e pular para outro documento. Quando você move o mouse sobre um link, a seta do mouse se transforma em uma pequena mão.

A tag HTML `<a>` define um hyperlink. O atributo mais importante do elemento `<a>` é o atributo `href`, que indica o destino do link. O texto do link é a parte que ficará visível para o leitor. Clicar no texto do link enviará o leitor para o endereço de URL especificado no `href`.

`<a href='https://www.exemplo.com'>clique aqui para ir para o exemplo</a>`



## Atributos de destino

Por padrão, a página vinculada será exibida na janela atual do navegador. Para alterar isso, você deve especificar outro destino para o link. O atributo **target** especifica onde abrir o documento vinculado, ele pode ter um dos seguintes valores:

- **\_self**- Predefinição. Abre o documento na mesma janela/guia em que foi clicado
- **\_blank**- Abre o documento em uma nova janela ou guia
- **\_parent** - Abre o documento no quadro pai
- **\_top** - Abre o documento em todo o corpo da janela

Exemplo: Use target="\_blank" para abrir o documento vinculado em uma nova janela ou guia do navegador:

```
<a href="https://www.w3schools.com/" target="_blank">Visit W3Schools!</a>
```

## Link para um endereço de e-mail

Use mailto:dentro do atributo "href" para criar um link que abre o programa de e-mail do usuário (para permitir que ele envie um novo e-mail):

```
<a href="mailto:someone@example.com">Send email</a>
```

## Títulos de links

O atributo title especifica informações extras sobre um elemento. As informações geralmente são mostradas como um texto de dica de ferramenta quando o mouse se move sobre o elemento.

```
<a href="https://www.w3schools.com/html/" title="Go to W3Schools HTML section">Visit our HTML Tutorial</a>
```

## IMAGENS

A tag HTML <img> é usada para incorporar uma imagem em uma página da web. As imagens não são tecnicamente inseridas em uma página da web; as imagens estão vinculadas a páginas da web. A tag <img> cria um espaço de retenção para a imagem referenciada e possui tem dois atributos obrigatórios:

- src - Especifica o caminho para a imagem
- alt - Especifica um texto alternativo para a imagem

### Mapas de imagem

A ideia por trás de um mapa de imagem é que você possa realizar ações diferentes dependendo de onde você clicar na imagem. Para criar um mapa de imagem, você precisa de uma imagem e de algum código HTML que descreva as áreas clicáveis. A tag HTML <map> define um mapa de imagem. As áreas são definidas com uma ou mais tags <area>.

Aqui está o código-fonte HTML para o mapa de imagem:

```


<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">
</map>
```

A imagem é inserida usando a <img> tag. A única diferença de outras imagens é que você deve adicionar um atributo usemap. O valor inserido em usemap começa com uma marca de hash #seguida pelo nome do mapa de imagem e é usado para criar um relacionamento entre a imagem e o mapa de imagem. Uma área clicável é definida usando um elemento <area>. Deve-se definir uma forma para a área clicável e escolher um dos seguintes valores:

- rect- define uma região retangular
- circle- define uma região circular
- poly- define uma região poligonal
- default- define toda a região

Por fim deve-se definir algumas coordenadas para poder especificar a área clicável na imagem.

## QUEBRA DE LINHA

A tag `<br>` insere uma única quebra de linha. A tag `<br>` é útil para escrever endereços ou poemas. A tag `<br>` é uma tag vazia, o que significa que não possui tag final. A tag `<br>` deve ser usada para inserir quebras de linha, não para adicionar espaço entre parágrafos.

Ex: (poema de Shakespeare)

`<p>Be not afraid of greatness.<br>`

`Some are born great,<br>`

`some achieve greatness,<br>`

`and others have greatness thrust upon them.</p>`

`<p><em>-William Shakespeare</em></p>`

## LISTAS

As listas HTML permitem que os desenvolvedores de Web agrupem em uma lista um conjunto de itens que possuem alguma relação.

### Lista HTML não ordenada `<ul>`

Uma lista não ordenada começa com a tag `<ul>`. Cada item da lista começa com a tag `<li>`. Os itens da lista serão marcados com marcadores (pequenos círculos pretos) por padrão. Ex: `<ul> <li>Café</li> <li>Chá</li><li>Leite</li></ul>`

- Café
- Chá
- Leite

### Lista HTML ordenada `<ol>`

Uma lista ordenada começa com a tag `<ol>`. Cada item da lista começa com a tag `<li>`. Os itens da lista serão marcados com números por padrão, definindo uma ordem específica para eles. Ex: `<ol> <li>Café</li> <li>Chá</li><li>Leite</li></ol>`

1. Café
2. Chá
3. Leite

### **Listas de descrição <dl>**

O HTML também suporta listas de descrição. Uma lista de descrição é uma lista de termos, com uma descrição de cada termo. A tag <dl> define a lista de descrição, a tag <dt> define o termo (nome) e a tag <dd> descreve cada termo. Ex: <dl> <dt>Café</dt> <dd>- bebida preta quente</dd> <dt>Leite</dt> <dd>- bebida branca gelada</dd> </dl>

Café

- bebida quente preta

Leite

- bebida branca gelada

## **TABELAS**

### **Definir uma tabela HTML <table>**

Uma tabela em HTML consiste em células de tabela dentro de linhas e colunas. A tag <table> Define uma tabela, onde ela começa e onde ela termina com a tag de fechamento</table>.

### **Linhas da tabela <tr>**

Cada linha da tabela começa com um <tr> e termina com uma tag </tr>. É possível ter quantas linhas quiser em uma tabela, mas também é necessário certifique-se de que o número de células seja o mesmo em cada linha.

### **Células da Tabela <td>**

Cada célula da tabela é definida por uma tag <td> e </td>. Tudo entre <td> e </td> é o conteúdo da célula da tabela.

## Cabeçalhos de tabela <th>

Para que as células sejam cabeçalhos, deve-se usar a tag <th>. Por padrão, o texto nos elementos <th> está em negrito e centralizado, mas você pode alterar isso com CSS.

Ex:

| Person 1 | Person 2 | Person 3 |
|----------|----------|----------|
| Emil     | Tobias   | Linus    |
| 16       | 14       | 10       |

## ELEMENTO <div>

A tag HTML <div>, que corresponde a abreviatura de Division, é um elemento HTML utilizado para dividir e agrupar os itens que serão exibidos em uma página. Por isso, ele funciona como um container, que pode receber diferentes tipos de conteúdo e organizá-los de várias maneiras.

### <div> e <span>

As tags <div> e <span> possuem funções parecidas, a principal diferença entre elas é o atributo display, que por padrão é definido como “block” na tag <div> e “inline” na tag <span>. O display block significa que existe uma quebra de linha no início e no encerramento da tag. Além disso, o elemento <div> ocupa toda a linha caso o seu tamanho não seja especificado, mesmo que seu conteúdo termine antes. Já a tag <span> ocupa somente o espaço necessário para a exibição dos elementos dentro dela.

## FORMULÁRIOS HTML

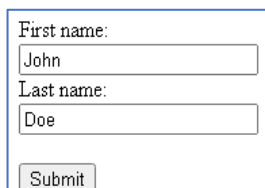
Um formulário HTML é usado para coletar o input de um usuário. Para criar um formulário é utilizado o elemento **<form>**. Esse elemento pode conter diferentes tipos de elementos input.

### Atributo action

O atributo action define a ação que será executada quando o formulário é enviado (submit).

No exemplo abaixo, os dados do formulário são enviados a um arquivo chamado “action\_page.php”. Esse arquivo contém um script no lado do servidor que trata esses dados:

```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="John"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="Doe"><br><br>  
  <input type="submit" value="Submit">  
</form>
```



Se o valor do atributo action não for especificado, o valor será definido para a página atual.

### Atributo method

O atributo method especifica o método de HTTP que será utilizado quando os dados de usuário forem enviados. Os dados de formulário podem ser enviados como uma variável da URL (com o “**method= “get”**”) ou como um post de transação HTTP (com o “**method= “post”**”).

Exemplos: Foi acrescentado o atributo method ao valor anterior com exemplos da utilização do valor get e do valor post.

```
<form action="/action_page.php" method="get">  
w3schools.com/action_page.php?fname=John&lname=Doe
```

```
<form action="/action_page.php" method="post">  
w3schools.com/action_page.php
```

**Method="get":**

- Anexa os dados de formulário à URL, em pares de nomes/valores.
- Nunca use GET para enviar dados sensíveis (o envio de dados é visível na URL).
- O tamanho de uma URL é limitado (2048 caracteres)
- Útil para enviar formulários em que um usuário quer registrar os resultados (para adicionar a URL aos favoritos, compartilhar a URL, registrá-la em algum documento, etc)
- GET é bom para dados que não estejam relacionados a segurança, como consulta de palavras no Google.

**Method="post":**

- Anexa os dados de formulário dentro do corpo da solicitação HTTP (o envio dos dados de formulário não é exibido na URL).
- POST não tem limite de tamanho, e pode ser usado para enviar grande quantidade de dados.
- Envio de formulários com POST não podem ser registrados.
- Sempre use POST para enviar dados sensíveis ou informações pessoais.

**Atributo target**

O atributo target especifica onde será exibida a resposta que foi recebida após o envio do formulário. Esse atributo pode ter um dos seguintes valores:

- Target="blank": A resposta será exibida em uma nova janela ou aba.
- Target="self": A resposta será exibida na janela atual.
- Target="parent": A resposta será exibida no quando pai.
- Target="top": A resposta será exibida no corpo inteiro da janela.
- Target="frameName": A resposta será exibida em um iframe nomeado.

Por padrão é atribuído o valor "\_self", o que significa que a resposta será exibida na janela atual.

### Atributo autocomplete

O atributo autocomplete especifica quando o formulário pode ser autocompletado. O navegador pode automaticamente completar os valores baseado em valores que o usuário inseriu antes.

```
<form action="/action_page.php" autocomplete="on">
```

First name:

Email:

### Atributo novalidate

O atributo novalidate é um atributo booleano. Quando presente, ele especifica que os dados dos formulários podem não ser validados quando enviados. Como exemplo, ele pode ser utilizado em um cadastro quando o usuário digita um número de telefone ou endereço de e-mail inválidos ou já registrado.

## ELEMENTOS DE FORMULÁRIO

O elemento <form> pode conter um ou mais dos seguintes elementos. É importante lembrar que todos os elementos de formulário devem ter o atributo “name” e serem nomeados para seus dados serem recebidos adequadamente pelo servidor.

### Elemento <input>

Um dos mais usados elementos de formulário é o elemento <input>. Ele poderá se comportar e ser exibido de diversas maneiras dependendo do tipo de atributo especificado nele.

### Elemento <label>

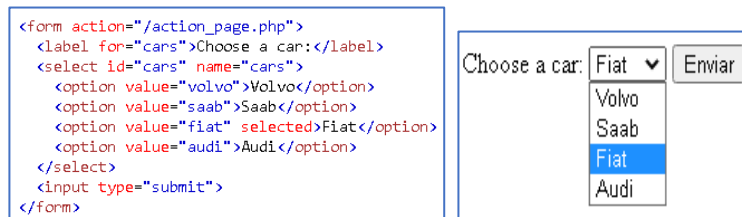
O elemento <label> define um rótulo para diversos elementos de formulário. É útil para usuários que utilizam leitores de tela e para acessibilidade, já que o leitor de tela pronunciará em voz alta o que o elemento focado pelo usuário. Esse elemento também pode ajudar usuários que tem dificuldade para clicar/interagir com pequenas regiões (como um radio buttons ou checkboxes),



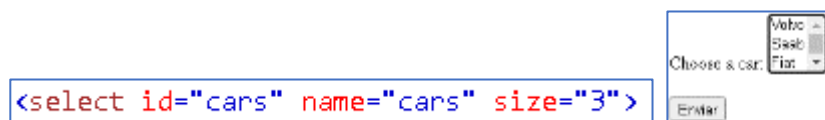
já que quando o usuário clica no texto dentro do elemento `<label>`, isso alterna a opção do radio boton/checkboxes.

### Elemento `<select>`

O elemento `<select>` define uma lista suspensa (drop-down list). O elemento `<option>` define uma opção que pode ser selecionada. Por padrão, o primeiro item na lista suspensa é selecionado. Para definir uma opção pré-selecionada, deve-se adicionar o atributo “selected” na opção desejada.

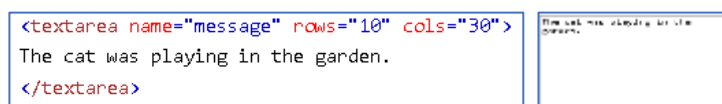


O atributo **size** especifica o numero de valores visíveis na lista suspensa. O atributo **multiple** permite que o usuário selecione mais de um valor na lista.



### Elemento `<textarea>`

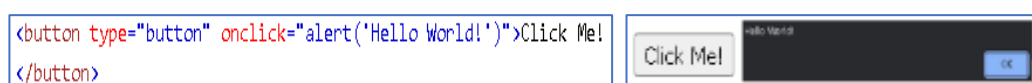
O elemento `<textarea>` especifica um campo de input com várias linhas (uma área de texto).



O atributo **rows** especifica o número de linhas visíveis na área de texto. O atributo **cols** especifica a largura visível da área de texto. Porém, o tamanho do elemento `textarea` geralmente alterado utilizando CSS.

### Elemento `<button>`

O elemento `<button>` define um botão clicável.



Sempre é importante especificar o atributo `type` do para um elemento `<button>`.

## Elementos <fieldset> e <legend>

O elemento <fieldset> é usado para agrupar um grupo de dados relacionados em um formulário. O elemento <legend> define uma descrição do elemento <fieldset>.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

## Elemento <datalist>

O elemento <datalist> especifica uma lista de opções pré-definidas para um elemento <input>. Usuários vão ver uma lista suspensa com opções predefinidas à medida que eles inserem os dados.

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

O atributo **list** especificado no elemento <input>, deve-se referir ao id do elemento <datalist>.

## Elemento <output>

O elemento <output> representa o resultado de um calculo (como o realizado por um script).

```
<form action="/action_page.php"
oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```

## TIPOS DE INPUT

### <input type="text">

Define um campo de texto em uma única linha, ele possui uma largura padrão de 20 caracteres.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

First name:

Last name:

### <input type="password">

Define um input com um campo para inserir senha.

```
<form>
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

Username:

romulo

Password:

••••••••

### <input type="submit">

Define um botão para envio dos dados de formulário. Os dados são enviados a um manipulador de formulário que geralmente é uma pagina de servidor com um script para processamento de dados enviados. O manipulador de formulário (form-handler) deve ser especificado no atributo **<action>**.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

First name:

John

Last name:

Doe

Submit

### <input type="reset">

Define um botão para redefinir todos os valores do formulário para os valores padrões.

```
<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" >
  <input type="reset">
</form>
```

First name:

Romulo

Last name:

Ramos

Enviar

Redefinir

First name:

John

Last name:

Doe

Enviar

Redefinir

### <input type="radio">

Define um botão de rádio onde o usuário pode selecionar somente uma opção dentre um número limitado.

```
<p>Choose your favorite Web language:</p>
<form action="/action_page.php">
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label><br><br>
  <input type="submit" value="Submit">
</form>
```

Choose your favorite Web language:

☐ HTML

☐ CSS

☒ JavaScript

### <input type="checkbox ">

Define uma caixa de seleção que permite que o usuário escolha nenhuma ou mais opções dentre um número limitado de escolhas.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

☒ I have a bike

☐ I have a car

☒ I have a boat

### <input type="button">

Define um input do tipo botão, esse botão pode receber uma função com scripts por exemplo.

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

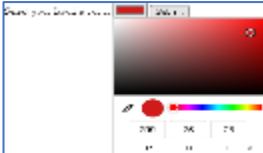
Click Me!

Hello World!

### <input type="color">

É usado para inserir um campo de seleção de cores. Alguns navegadores não suportam esse tipo de input.

```
<form>
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor">
</form>
```



### <input type="date ">

É usado para inserir um campo de seleção de datas. Nem todos os navegadores dão suporte para esse tipo de input. É possível alterar e restringir as datas limite de seleção com o uso dos atributos **max** e **min**.

```

<form>
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
</form>

```



### **<input type="datetime-local">**

É usado para inserir um campo de seleção de data e hora. Nem todos os navegadores dão suporte para esse tipo de input.

### **<input type="month">**

É usado para inserir um campo de seleção de data para somente mês e ano.

### **<input type="week">**

É usado para inserir um campo de seleção de data para somente semana e ano.

### **<input type="time">**

É usado para inserir um campo de seleção de horas ( sem fuso horário).

### **<input type="email ">**

É usado em campos de input que podem conter endereços de e-mail. Dependendo do suporte do navegador a validação automática quando o e-mail é enviado pode não funcionar.

### **<input type="file ">**

Define um campo de seleção de uploads de arquivos.

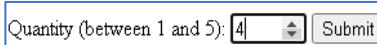
### **<input type="hidden ">**

Define um campo de input oculto (que não é visível ao usuário). Um campo oculto permite que desenvolvedores de web possam incluir dados que não podem ser vistos ou modificados quando o formulário é enviado. Apesar do valor não ser exibido para o usuário comum ainda é possível visualiza-lo através de ferramentas de desenvolvedor ou da exibição do código fonte, por isso não deve ser utilizado para lidar com dados sensíveis.

### **<input type="number">**

Define um campo para inserir números. É possível alterar algumas restrições dos números que serão aceitos. O seguinte exemplo mostra um campo de input de número onde só é permitido inserir um número de 1 a 5:

```
<form>
  <label for="quantity">Quantity (between 1 and 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```



### **<input type="range">**

Define uma barra deslizante com um intervalo para seleção de números (utilizado para quando o valor exato do número não é muito importante). Por padrão o intervalo é de 0 a 100 mas pode ser alterado com e acrescentadas diversas restrições

### **<input type="search">**

É usado para inserir um campo de busca (um campo de busca se comporta como um campo de texto normal).

### **<input type="tel">**

É usado para inserir um campo de input que pode conter um número de telefone.

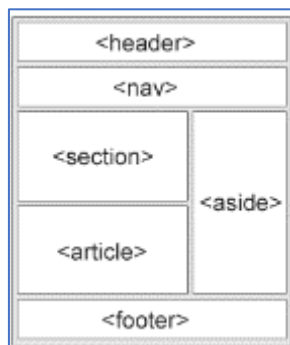
### **<input type="url">**

É usado para inserir um campo de input que pode conter um ENDEREÇO URL. Dependendo do suporte do navegador, a url pode ser automaticamente validada quando enviada.



## ELEMENTOS SEMÂNTICOS

Um elemento semântico descreve de maneira clara o significado/sentido de elementos tanto para o navegador quanto para o desenvolvedor. As tags semânticas podem ser usadas para definir diferentes partes de uma página de web tornando a navegação mais acessível e facilitando mecanismos de busca.



### ELEMENTO <section>

Define uma seção em um documento. Uma seção é um agrupamento temático de conteúdo, normalmente com um título. Pode ser usado para separar capítulos, introdução, novos itens, informação de contato. Uma página web normalmente pode ser separada em seções para introdução, conteúdo e informações de contato.

### ELEMENTO <article>

Especifica um artigo, conteúdo autônomo, independente, etc. Ele não necessita de outros conteúdos para fazer sentido, e pode ser distribuído de maneira independente no restante da página web. O elemento <article> pode ser usado em: posts de fórum, posts de blogs, comentários de usuários, cartões de produtos, artigos de notícias, etc.

### ELEMENTO <header >

Representa um conteúdo introdutório ou o título do conteúdo. Esse elemento normalmente contém: um ou mais elementos heading (<h1> - <h6>), logo ou ícones, informações de autoria.

### ELEMENTO <footer>

Define um rodapé para um documento ou uma seção. Ele geralmente contém: informações de autoria, informações sobre direitos autorais,

informações de contato, mapa do site, links para retornar ao topo da página, documentos/páginas relacionadas.

### ELEMENTO <nav>

Define uma barra de navegação que contém os principais links necessários para a navegação na página.

### ELEMENTO <aside>

Define o conteúdo posicionado nas laterais (como uma barra de slide). Esse elemento deve estar indiretamente relacionado com o conteúdo do entorno.

### ELEMENTOS <figure> e <figcaption>

O elemento <figure> especifica conteúdos independentes como ilustrações, diagramas, fotos, gráficos, linhas de código, etc. O elemento <figcaption> define uma descrição ou legenda para um elemento <figure>.

## HTML ATRIBUTOS PERSONALIZADAS

Os atributos dos elementos HTML providenciam informações adicionais sobre os elementos, eles são sempre especificados no começo da tag e normalmente vêm em pares com nome/valor (como: name = "value"). O atributo "href" por exemplo, especifica o endereço URL em que o link deve ir.

```
<a href="https://www.w3schools.com">Visit W3Schools</a> Visit W3Schools
```

Através do uso de métodos DOM é possível acessar alterar e criar esses atributos. Através do método **getAttribute()**, é possível acessar o valor de atributo específico do elemento HTML e o seu valor. De maneira semelhante, o método **setAttribute()** permite a implementação de um novo atributo no elemento.

No seguinte exemplo é atribuído a uma variável o valor do atributo class de um elemento.

```
<h1 id="myH1" class="democlass">The Element Object</h1>
<p id="demo"></p>
<script>
const element = document.getElementById("myH1");
let text = element.getAttribute("class");
document.getElementById("demo").innerHTML = text;
</script>
```

**The Element Object**

democlass



No seguinte exemplo é criada um estilo para uma classe específica, essa classe é um atributo que, após o evento “click” ocorrer ela é adicionada a um elemento HTML mudando então o seu estilo.



Essa é uma maneira de acessar e alterar atributos de elementos HTML através do DOM, porém, quando se trata de atributos personalizados, é mais prático e correto a utilização da propriedade `HTMLElement.dataset`. A propriedade `dataset` permite acessar e alterar atributos personalizados de um elemento, ou seja, atributos não usuais e que não são utilizados como padrão, eles são chamados de atributos globais `data-*`, eles formam uma classe de atributos conhecida como `custom data attributes`, que permite que informações proprietárias sejam trocadas via script entre o HTML e sua representação DOM.

A propriedade `dataset` é um recurso muito interessante, permite que adicionemos dados a elementos HTML, como se fossem constantes, por exemplo podemos criar um elemento e definimos uma série de atributos que funcionarão como informações deste elemento, que pode ser facilmente obtido por javascript. Veja um exemplo simples, onde adicionamos três dados `vel`, `pot` e `cor`, note que basta adicionar “`data-`” antes do nome do atributo.

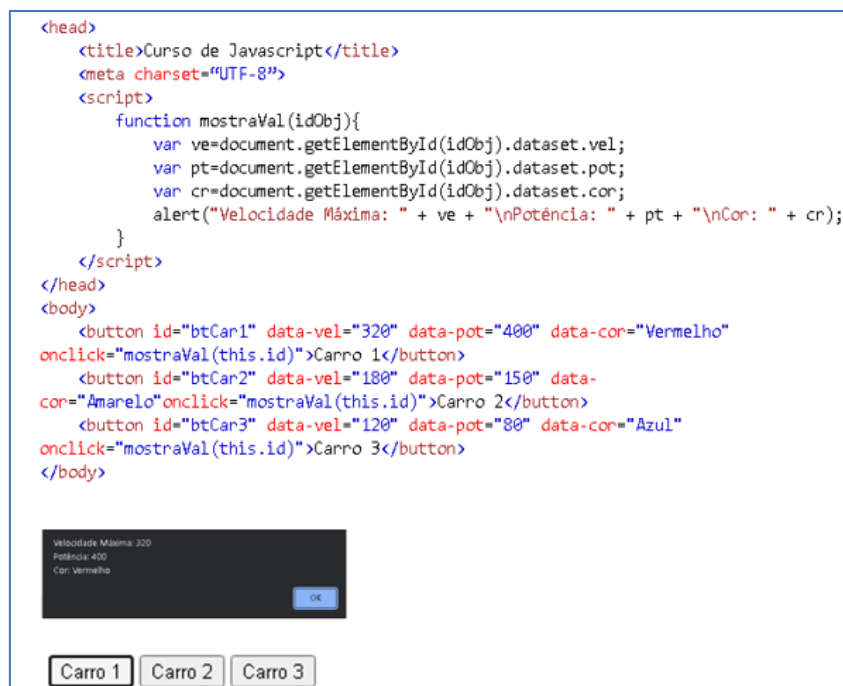
```
<button id="btCar1" data-vel="320" data-pot="400" data-cor="Vermelho">
```

Para pegar os valores usamos a seguinte sintaxe.

```
element.dataset.keyname
```

```
var ve=document.getElementById(idObj).dataset.vel;
var pt=document.getElementById(idObj).dataset.pot;
var cr=document.getElementById(idObj).dataset.cor;
```

Em nosso programa de exemplo criamos três botões com três atributos cada e uma função para pegar o valor destes atributos e mostrar em um alert.



## HTML MULTIMÍDIA

Multimídia vem em muitos formatos diferentes. Pode ser quase qualquer coisa que você possa ouvir ou ver, como imagens, música, som, vídeos, discos, filmes, animações e muito mais. As páginas da Web geralmente contêm elementos multimídia de diferentes tipos e formatos.

### HTML VIDEO

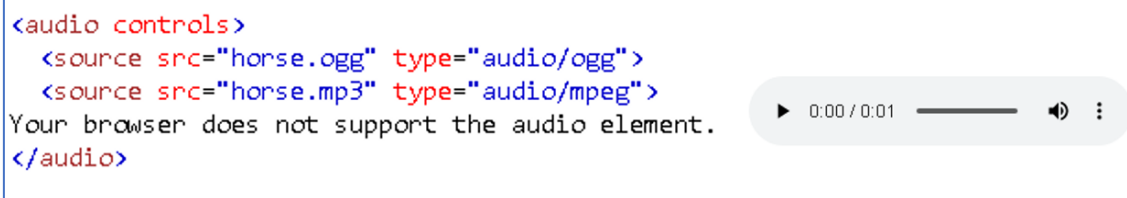
O elemento `<vídeo>` é usado para exibir um vídeo na página do navegador, esse elemento possui uma tag de abertura e de fechamento, o endereço da mídia de vídeo pode tanto ser especificada como um tributo do elemento `"scr"` ou como um elemento `<source src="vídeo.mp4" type="vídeo/mp4">` dentro do elemento vídeo, como no exemplo.



O atributo `"control"` adiciona botões para controle de vídeo, como play, pause e volume.

## HTML AUDIO

O elemento HTML <audio> é usado para reproduzir um arquivo de áudio em uma página da web.



O atributo controls adiciona controles de áudio, como reproduzir, pausar e volume. O elemento <source> permite especificar arquivos de áudio alternativos que o navegador pode escolher. O navegador usará o primeiro formato reconhecido. O texto entre as tags <audio> e </audio> só será exibido em navegadores que não suportam o elemento <audio>.

## PROPRIEDADES AUDIO/VÍDEO

| PROPRIEDADE                       | DESCRIÇÃO  |
|-----------------------------------|--|
| <b><u>audioTracks</u></b>         | Retorna um objeto AudioTrackList representando uma faixa de áudio disponível                         |
| <b><u>autoplay</u></b>            | Define ou retorna se o áudio/vídeo deve ser reproduzido automaticamente quando carregado.            |
| <b><u>buffered</u></b>            | Retorna um objeto TimeRanges representando as partes do áudio/vídeo que estão armazenadas em buffer. |
| <b><u>controller</u></b>          | Retorna o objeto MediaController representando o controlador da mídia atual.                         |
| <b><u>controls</u></b>            | Define ou retorna se o áudio/vídeo deve exibir os controles de reprodução (como play, pause, etc.)   |
| <b><u>crossOrigin</u></b>         | Define ou retorna as configurações CORS do áudio/vídeo   |
| <b><u>currentSrc</u></b>          | Retorna a URL do áudio/vídeo atual   |
| <b><u>currentTime</u></b>         | Define ou retorna a posição de reprodução atual do áudio/vídeo (em segundos)                         |
| <b><u>defaultMuted</u></b>        | Define ou retorna se o vídeo deve estar mutado por padrão  |
| <b><u>defaultPlaybackRate</u></b> | Define ou retorna a velocidade padrão de reprodução do áudio/vídeo                                   |
| <b><u>duration</u></b>            | Retorna o tempo de duração do áudio/vídeo (em segundos)  |
| <b><u>ended</u></b>               | Retorna se a reprodução do áudio/vídeo deve ter fim ou não   |
| <b><u>error</u></b>               | Retorna um objeto MediaError representando o estado do erro do áudio/vídeo                           |
| <b><u>loop</u></b>                | Define ou retorna se o áudio/vídeo deve começar novamente após ele ter terminado.                    |
| <b><u>mediaGroup</u></b>          | Define ou retorna o grupo do qual a mídia áudio/vídeo pertence.                                      |
| <b><u>muted</u></b>               | Define ou retorna se o vídeo está mutado ou não  |
| <b><u>networkState</u></b>        | Retorna o estado atual da rede do áudio/vídeo  |
| <b><u>paused</u></b>              | Retorna se o áudio/vídeo está pausado ou não   |
| <b><u>playbackRate</u></b>        | Define ou retorna a velocidade de reprodução   |

|                           |  |
|---------------------------|--|
| <b><u>played</u></b>      | Retorna um objeto TimeRanges representando as partes do áudio/vídeo que já foram reproduzidas. |
| <b><u>preload</u></b>     | Define ou retorna se o áudio/vídeo deve ser carregado durante o carregamento da página         |
| <b><u>readyState</u></b>  | Retorna o estado de carregamento do áudio/vídeo  |
| <b><u>seekable</u></b>    | Retorna um objeto TimeRanges representando as partes "seekable" do áudio/vídeo                 |
| <b><u>seeking</u></b>     | Retorna se o usuário está atualmente "seeking" no áudio/vídeo                                  |
| <b><u>src</u></b>         | Define ou retorna a fonte do elemento audio/vídeo  |
| <b><u>startDate</u></b>   | Retorna um objeto Data representando o deslocamento de tempo atual                             |
| <b><u>textTracks</u></b>  | Retorna um objeto TextTrackList representando as faixas de texto disponíveis                   |
| <b><u>videoTracks</u></b> | Retorna um objeto VideoTracklist representando as faixas de vídeo disponíveis                  |
| <b><u>volume</u></b>      | Define ou retorna o volume do áudio/vídeo  |

## EVENTOS AUDIO/VÍDEO

| EVENTOS                      | DESCRIÇÃO   |
|------------------------------|---|
| <b><u>abort</u></b>          | Dispara quando o carregamento do áudio/vídeo é interrompido                                       |
| <b><u>canplay</u></b>        | Dispara quando o navegador pode começar a reproduzir a mídia                                      |
| <b><u>canplaythrough</u></b> | Dispara quando o navegador pode reproduzir a mídia sem parar para armazenar em buffer             |
| <b><u>durationchange</u></b> | Dispara quando a duração da mídia é alterada  |
| <b><u>emptied</u></b>        | Dispara quando a playlist atual está vazia  |
| <b><u>ended</u></b>          | Dispara quando a playlist atual termina   |
| <b><u>error</u></b>          | Dispara quando um erro ocorre durante o carregamento da mídia                                     |
| <b><u>loadeddata</u></b>     | Dispara quando o navegador carregou o frame atual da mídia  |
| <b><u>loadedmetadata</u></b> | Dispara quando o navegador carregou os metadados da mídia   |
| <b><u>loadstart</u></b>      | Dispara quando o navegador começa a "carregar" a mídia  |
| <b><u>pause</u></b>          | Dispara quando a mídia é pausada  |
| <b><u>play</u></b>           | Dispara quando a mídia começa a ser reproduzido   |
| <b><u>playing</u></b>        | Dispara quando a mídia está sendo reproduzida após ter sido pausada ou interrompida por buffering |
| <b><u>progress</u></b>       | Dispara quando o navegador está realizando downloading da mídia                                   |
| <b><u>ratechange</u></b>     | Dispara quando a velocidade de reprodução é alterada  |
| <b><u>seeked</u></b>         | Dispara quando o usuário termina de mover/pular para uma nova posição no áudio/vídeo              |
| <b><u>seeking</u></b>        | Dispara quando o usuário começa a mover/pular para uma nova posição no áudio/vídeo                |
| <b><u>stalled</u></b>        | Dispara quando o navegador está tentando obter os dados da mídia, mas eles não estão disponíveis  |
| <b><u>suspend</u></b>        | Dispara quando o navegador intencionalmente está tentando obter os dados da mídia                 |
| <b><u>timeupdate</u></b>     | Dispara quando o tempo de reprodução atual é alterado   |
| <b><u>volumechange</u></b>   | Dispara quando o volume é alterado  |
| <b><u>waiting</u></b>        | Dispara quando o vídeo é pausado por precisar carregar o próximo frame                            |

## MÉTODOS AUDIO/VÍDEO

| Método                       | Descrição   |
|------------------------------|---|
| <b><u>addTextTrack()</u></b> | Adiciona um novo text track ao áudio/vídeo                                |
| <b><u>canPlayType()</u></b>  | Verifica se o navegador pode reproduzir um tipo específico de áudio/vídeo |
| <b><u>load()</u></b>         | Recarrega o elemento áudio/vídeo  |
| <b><u>play()</u></b>         | Começa a reproduzir o áudio/vídeo   |
| <b><u>pause()</u></b>        | Pausa a reprodução do áudio/vídeo   |

## IFRAME

Um iframe HTML é usada para exibir uma página web dentro de outra página web. Um iframe é escrito utilizando a tag <iframe> que especifica um frame “inline” (em linha).

```
<iframe src="url" title="description"></iframe>
```

O seguinte exemplo inclui um vídeo do Youtube dentro de um site HTML com o uso do iframe, é possível ainda observar alguns atributos desse iframe, como o “src” que especifica o endereço web desse vídeo, o atributo “title” que determina um nome para esse elemento iframe, “frameborder” que determina o tipo de borda do iframe, “allow” que especifica uma série de recursos permitidos dentro desse iframe, e por último um atributo “allowfullscreen” que permite esse iframe ser exibido em tela cheia.

```
<body>
<h2>Iframe - Target for a Link</h2>
<iframe width="560" height="315"
src="https://www.youtube.com/embed/SOxutBMCOUc" title="YouTube
video player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
</body>
</html>
```



## ATRIBUTOS IFRAME

O elemento <iframe> também suporta os Atributos Globais em HTML e possui alguns atributos próprios.

| Atributo                     | Valor   | Descrição  |
|------------------------------|---|--|
| <b>allow</b>                 |   | Especifica uma política de recurso para o elemento <iframe>  |
| <b>allowfullscreen</b>       | true<br>false   | Define se o <iframe> pode, ou não, ser visualizado em tela cheia.  |
| <b>allowpaymentrequest</b>   | true<br>false   | Define se o <iframe> pode, ou não, se o iframe pode solicitar um API de pagamento terceira.  |
| <b><u>height</u></b>         | <i>pixels</i>   | Especifica a altura do <iframe>. Por padrão ela é 150 pixels.  |
| <b>loading</b>               | eager<br>lazy   | Especifica como o navegador deve carregar o <iframe>   |
| <b><u>name</u></b>           | <i>text</i>   | Especifica o nome para identificação do <iframe>   |
| <b><u>referrerpolicy</u></b> | no-referrer<br>no-referrer-when-downgrade<br>origin<br>origin-when-cross-origin<br>same-origin<br>strict-origin-when-cross-origin<br>unsafe-url | Especifica quais informações de referências podem ser enviadas para o iframe, como coleta de dados para exibição de propaganda, etc. |
| <b><u>sandbox</u></b>        | allow-forms<br>allow-pointer-lock<br>allow-popups<br>allow-same-origin<br>allow-scripts<br>allow-top-navigation                                 | Permite a definição de restrições do conteúdo do iframe para controle de segurança.  |
| <b><u>src</u></b>            | <i>URL</i>  | Especifica o endereço do documento que será incorporado  |
| <b><u>srcdoc</u></b>         | <i>HTML_code</i>  | Especifica o conteúdo HTML da página incorporada   |
| <b><u>width</u></b>          | <i>pixels</i>   | Especifica a largura do <iframe>. Por padrão é 300 pixels.   |

## HTML CANVAS

O elemento `<canvas>` é usado em desenhos gráficos na página web com JavaScript. O elemento é apenas um container para esses desenhos gráficos, (uma tradução para canvas seria tela de desenho ou quadro), é necessário utilizar um script para realmente desenhar nessa “tela”.

Devem ser especificados alguns atributos no elemento canvas, como o “id” para ele ser referenciado corretamente pelo JavaScript. **Os atributos de largura e altura devem ser especificados no elemento HTML** e não como atributos de estilo no CSS por questões de compatibilidade com as coordenadas e os métodos canvas.

```
<canvas id="myCanvas" width="200"
height="100"
style="border:1px solid #000000;">
</canvas>
```



## DESENHANDO NO CANVAS COM JS

Para desenhar no canvas primeiro o elemento deve ser “encontrado” através do método HTML DOM **getElementById()**.

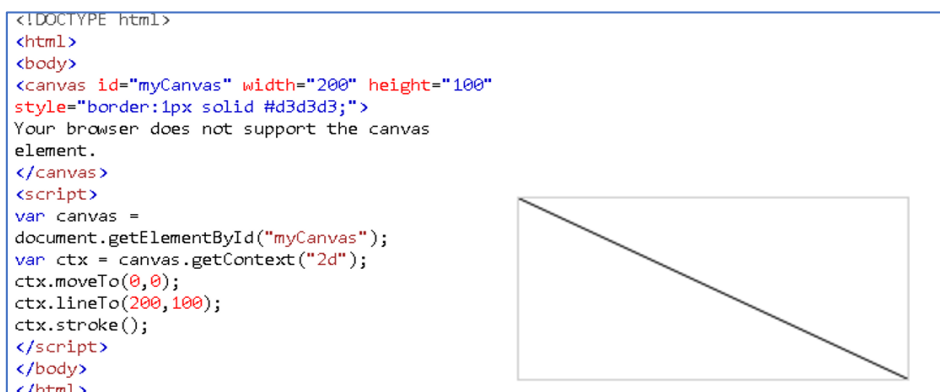
```
var canvas = document.getElementById("myCanvas");
```

Após isso é preciso especificar o tipo de objeto que será desenhado, o método **getContext()** é um objeto embutido no HTML com propriedades e métodos específicos para o desenho. É necessário especificar se o objeto deve ser em duas dimensões ou em três dimensões. Esse objeto é atribuído a uma variável.

```
var ctx = canvas.getContext("2d");
```

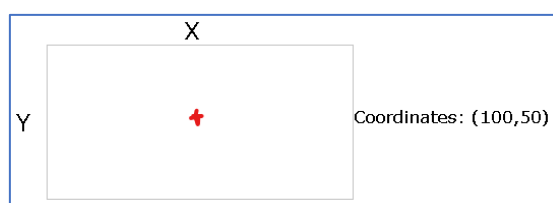
E finalmente após isso é possível então desenhar no canvas, no seguinte exemplo foi desenhada uma linha através dos métodos **moveTo()**, que especifica a posição do “pincel” com o uso de coordenadas relacionadas como tamanho em pixels da caixa da tela do elemento `<canvas>`, o método **lineTo()** especifica o “caminho” de uma linha até uma outra coordenada e por fim o

método **stroke()** desenha esse “caminho” (uma tradução para stroke seria “traçar” ou seja “traçar uma linha”).



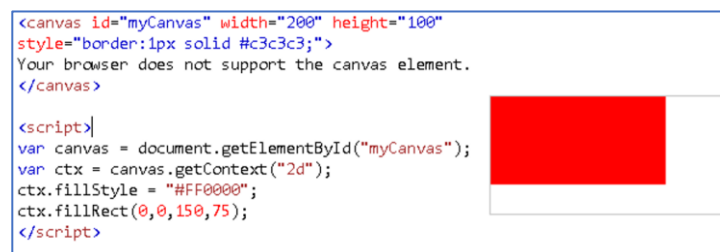
## COORDENADAS CANVAS

A tela HTML é uma grade bidimensional. O canto superior esquerdo da tela tem as coordenadas (0,0). O canto inferior direito da tela tem as coordenadas referentes ao tamanho da tela (no exemplo anterior 200,100).



## RETÂNGULO CANVAS

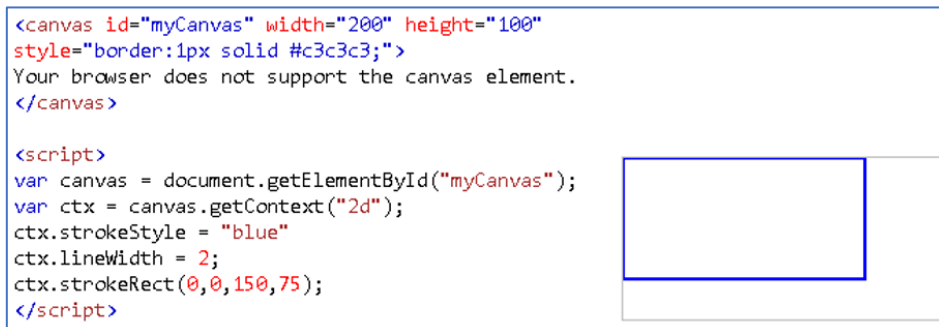
Para desenhar um retângulo uma tela canvas, é necessário utilizar os seguintes métodos e propriedades: **fillStyle** - propriedade que define a cor que deve preencher o desenho. **fillRect(x,y, largura, altura)** – método que desenha um retângulo preenchido pela cor antes definida.



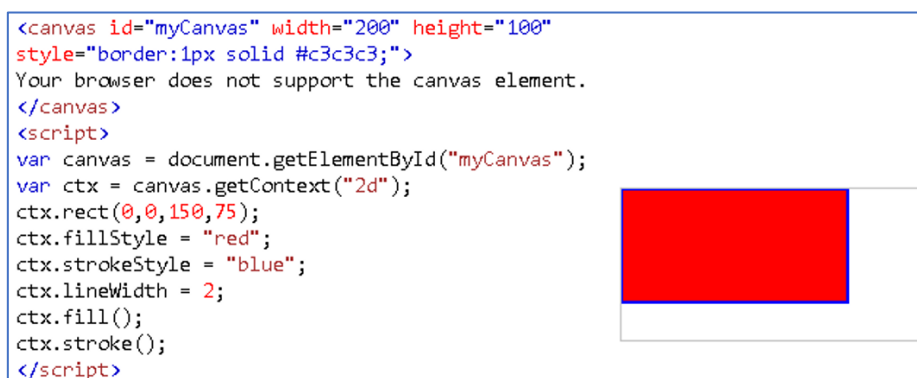
O exemplo acima demonstra o uso do canvas para o desenho de um retângulo preenchido, porém é possível também desenhar um retângulo sem preenchimento. No exemplo abaixo foi utilizada a propriedade **strokeStyle** para



definir a cor do traçado, a propriedade `lineWidth` define a espessura desse traçado e por fim o método **`strokeRect()`** desenha esse retângulo.

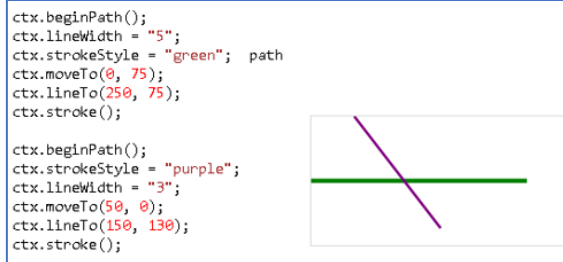


É possível também utilizar o método **`rect()`** para definir o ponto de início do retângulo e suas dimensões para desenhá-lo no final. O seguinte exemplo define um retângulo que começa nas coordenadas (0,0) e possui uma largura de 150 pixels e altura de 75 pixels, a propriedade `fillStyle` define o estilo do preenchimento dele e a propriedade `strokeStyle` define o estilo do traçado, a propriedade `lineWidth` define a largura desse traçado, e por fim, os métodos `fill()` e `stroke()` desenharam esse retângulo.

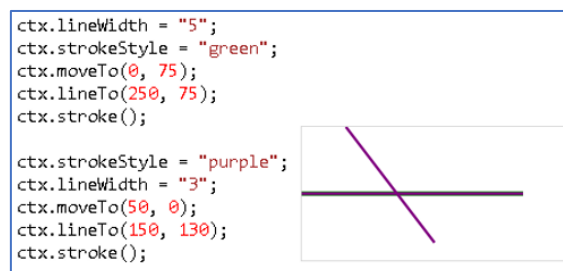


## PATH CANVAS

O método **`beginPath()`** inicia um caminho ou redefine o caminho atual. O ideal quando se utiliza o canvas é sempre utilizar esse método pois ele marca o começo do desenho atual possibilitando começar outro atribuindo diferentes propriedades para esse novo desenho. No seguinte exemplo foram desenhadas duas linhas com dois caminhos diferentes na tela, uma possui a cor roxa e a outra uma cor verde. Sem o uso do `beginPath()` as propriedades do traçado `strokeStyle` e `lineWidth` por exemplo são atribuídas a outra linha.



O seguinte exemplo demonstra como seria o resultado caso o método não fosse corretamente utilizado.



## CÍRCULO CANVAS

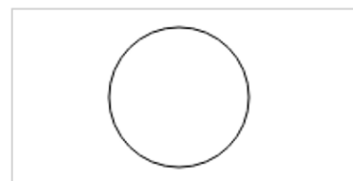
Para desenhar um círculo em uma tela canvas, são utilizados os seguintes métodos:

- **beginPath()** - inicia um caminho
- **arc(x, y, r, anguloInicial, anguloFinal)** - cria um arco/curva. Para criar um círculo com `arc()`: Deve ser definido o ângulo (em radianos) inicial como 0 e o ângulo final (em radianos) como  $2 * \text{Math.PI}$ . Os parâmetros x e y definem as coordenadas x e y do centro do círculo. O parâmetro r define o raio do círculo.
- **stroke()** - desenhar o traçado do círculo

```

<canvas id="myCanvas" width="200" height="100"
style="border:1px solid #d3d3d3;">
Your browser does not support the canvas element.
</canvas>
<script>
var canvas = document.getElementById("myCanvas");
var ctx = canvas.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
</script>

```



## IMAGENS CANVAS

Para desenhar uma imagem em uma tela canvas, deve ser utilizado o seguinte método:

- `drawImage( imagem,x,y )`

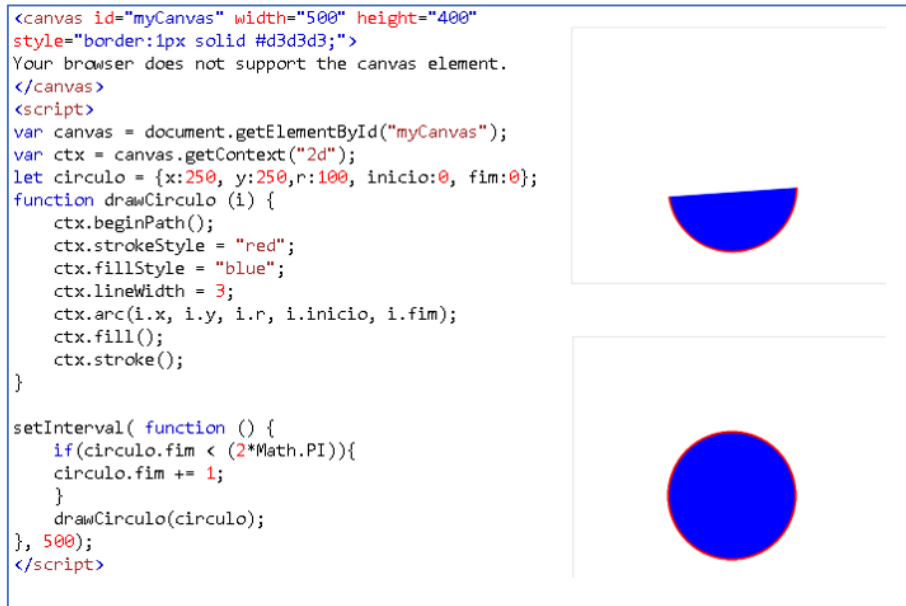


No exemplo acima uma imagem que não está sendo exibida (`display:none;`) é atribuída a uma variável no script com o uso do DOM, quando o evento “onload” dessa imagem é disparado, uma função é chamada, essa função faz o desenho dessa imagem no canvas.

## ANIMAÇÕES CANVAS

É canvas possui muitas utilidades desde do simples desenho até a criação de animações complexas, jogos interativos, aplicações, etc. Uma maneira de criar animações com o canvas é através do uso dos controles de intervalo. No seguinte exemplo, foi criado um objeto chamado de círculo, esse objeto possui suas propriedades que podem ser usadas como parâmetros para especificar as coordenadas x e y de um círculo, uma propriedade r referente ao parâmetro do raio, e propriedades referentes ao ângulo de início e fim do círculo. A função criada chamada de “`drawCirculo()`” desenha um círculo com essas propriedades especificadas no objeto círculo, porém importante observar que o ângulo final do círculo está definido como 0, ou seja esse círculo não pode ser desenhado pois não possui um valor final. Através do método de controle de intervalo **`setInterval()`** foi criada uma função que vai acrescentar 1 radianos ao valor do ângulo se uma condição for atendida, e vai invocar a função de desenhar o círculo, ou seja a função vai desenhar o círculo a cada 1 radiano e essa função

vai se repetir em um intervalo de 500 milissegundos conforme especificado no método, isso vai criar uma animação de desenho do círculo.



## Referências:

- <https://www.w3schools.com>
- <https://www.alura.com.br>
- <https://blog.betrybe.com>