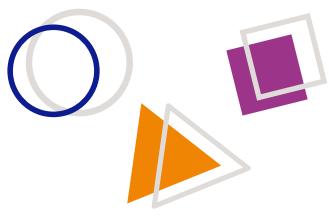




Mundo Tech



# PROGRAMAÇÃO FRONT-END ELEMENTOS IMPORTANTES

# SUMÁRIO

HTML5.....	3
Listas .....	4
Tabelas.....	5
Recursos de imagem, áudio e vídeo.....	6
Estrutura do DOM .....	8
Formulários .....	10
Referências.....	19



## MAS, ANTES DE COMEÇAR...

Para resolver seu desafio, você poderá utilizar qualquer imagem da internet e uma logo com a qual você mais se identifique. Para procurar imagens, acesse o site a seguir ou aproxime seu celular do QR code:

<https://www.istockphoto.com/br>



E, lembre-se da sugestão do vídeo: se você se sentir pouco inspirado para preencher linhas de textos, pode usar o Lorem Ipsum, clicando no link a seguir ou aproximando seu celular do QR code:



<https://loremipsum.io>

Boa sorte com seu primeiro desafio!

## HTML5

Como você conferiu, o HTML5 é um esforço para trazer ordem aos desafios do desenvolvimento web, organizando práticas comuns e adotando implementações de vários navegadores. HTML5 é simplesmente um termo genérico para a próxima geração de aplicativos da web e como a funcionalidade será expandida com melhor marcação (HTML), melhor estilo (CSS) e melhor interatividade (JavaScript). Dando continuidade aos estudos, agora você vai conferir alguns conceitos importantes relacionados a diversos tópicos que são necessários durante a construção de páginas web. Você compreenderá o processo de criação de listas e tabelas. Você também acompanhará os procedimentos para inserção de recursos multimídia, como imagem, áudio e vídeo. E, na sequência, conhecerá a estrutura do DOM (*Document Object Model*). Por fim, irá trabalhar com formulários, identificando os tipos de campos que podem ser utilizados em páginas web para coletar diversos tipos de informações. Vamos lá! Iniciaremos pelas listas.

# Listas

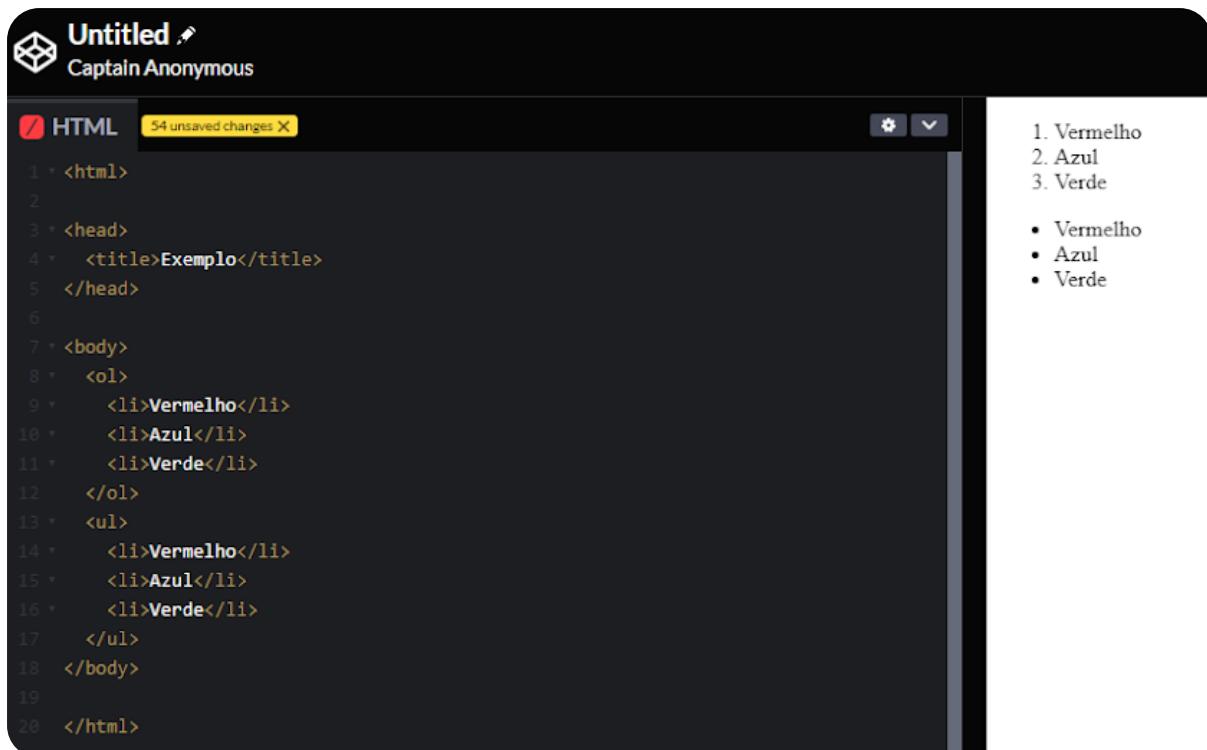
As listas são usadas para agrupar itens semelhantes. O HTML fornece dois tipos de tags de lista: listas ordenadas <ol> e não ordenadas <ul>.

Tag	Propósito
<ol>...</ol>	Cria uma lista numerada por padrão.
<ul>...</ul>	Cria uma lista com marcadores por padrão.
<li>...</li>	Indica um item de lista para lista ordenada e não ordenada.

Quadro 1 - Listas

Fonte: do Autor (2022)

Entenda com o exemplo:



The screenshot shows a code editor window titled "Untitled" with the subtitle "Captain Anonymous". The file type is "HTML" and it has "54 unsaved changes". The code is as follows:

```
1 <html>
2
3 <head>
4   <title>Exemplo</title>
5 </head>
6
7 <body>
8   <ol>
9     <li>Vermelho</li>
10    <li>Azul</li>
11    <li>Verde</li>
12  </ol>
13  <ul>
14    <li>Vermelho</li>
15    <li>Azul</li>
16    <li>Verde</li>
17  </ul>
18 </body>
19
20 </html>
```

To the right of the code editor, there are two lists generated from the code:

- 1. Vermelho
- 2. Azul
- 3. Verde

- Vermelho
- Azul
- Verde

Figura 1 - Estruturação de listas em HTML

Fonte: do Autor (2022)

# Tabelas

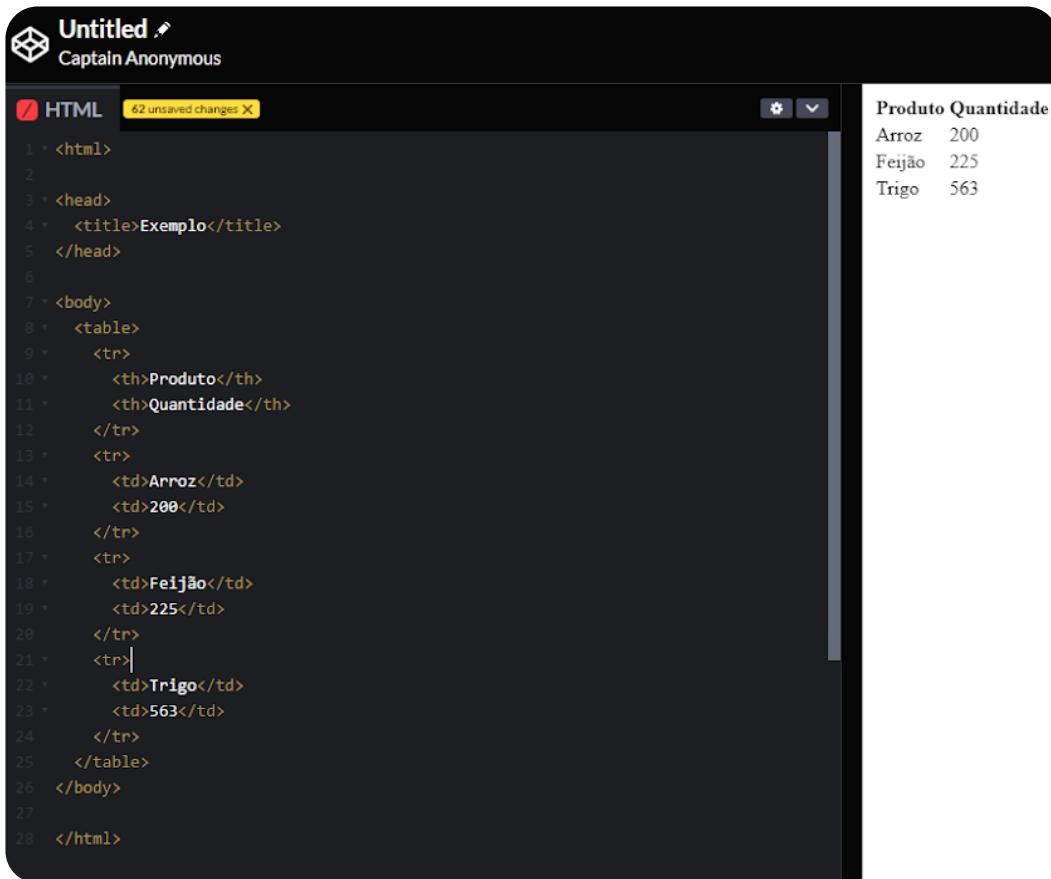
Uma tabela HTML é usada para organizar os dados em linhas e colunas, ou possivelmente em uma estrutura mais complexa. Uma tabela HTML é definida com a tag <table>. A seguir estão algumas tags básicas usadas para especificar os recursos de uma tabela:

Tag	Propósito
<table>...</table>	Para definir uma estrutura de tabela.
<th>...</th>	Para definir o cabeçalho da tabela.
<tr>...<tr>	Para definir a linha.
<td>...</td>	Para definir os dados da tabela ou uma célula.

Quadro 2 - Tabelas

Fonte: do Autor (2022)

Por padrão, os títulos das tabelas estão em negrito e centralizados, enquanto os dados não recebem negrito e ficam alinhados à esquerda. Acompanhe, a seguir, um exemplo:



The screenshot shows a code editor window titled "Untitled" with the subtitle "Captain Anonymous". The tab bar indicates it's an "HTML" file with "62 unsaved changes". The code editor displays the following HTML code:

```
1 <html>
2
3 <head>
4   <title>Exemplo</title>
5 </head>
6
7 <body>
8   <table>
9     <tr>
10    <th>Produto</th>
11    <th>Quantidade</th>
12  </tr>
13  <tr>
14    <td>Arroz</td>
15    <td>200</td>
16  </tr>
17  <tr>
18    <td>Feijão</td>
19    <td>225</td>
20  </tr>
21  <tr>
22    <td>Trigo</td>
23    <td>563</td>
24  </tr>
25 </table>
26 </body>
27
28 </html>
```

To the right of the code editor, there is a preview pane showing the rendered table with the following data:

Produto	Quantidade
Arroz	200
Feijão	225
Trigo	563

Figura 2 - Apresentação de uma tabela em HTML

Fonte: do Autor (2022)

# Recursos de imagem, áudio e vídeo

A tag <img> permite incorporar uma imagem em uma página web. Ele serve como um marcador de posição. Os formatos GIF, JPG e PNG são os compatíveis com imagens em um documento HTML. Confira um descriptivo de seus principais atributos:

Atributo	Propósito	Valor
alt (obrigatório)	Menciona o texto a ser exibido se a imagem não for exibida por algum motivo.	texto
src (obrigatório)	Menciona o caminho da imagem.	URL
height	Menciona a altura da imagem.	pixels
width	Menciona a largura da imagem.	pixels

Quadro 3 - Principais atributos em imagem, áudio e vídeo

Fonte: do Autor (2022)

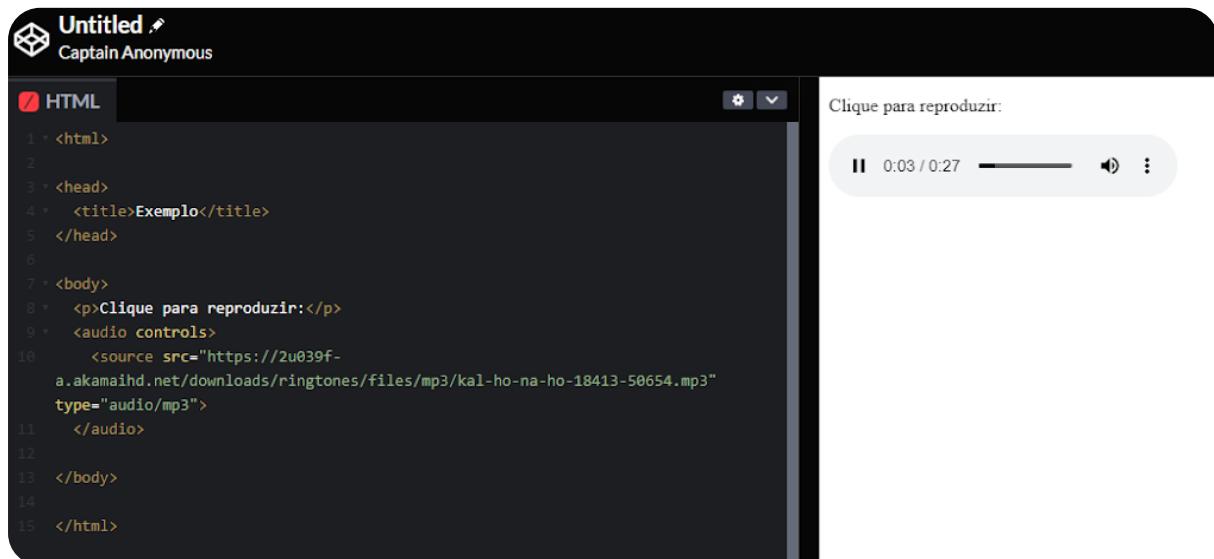
Compreenda mais com o exemplo:



Figura 3 - Elemento de imagem em HTML

Fonte: do Autor (2022)

Além disso, o HTML5 fornece uma maneira rápida e fácil de adicionar arquivos de áudio e vídeo para serem reproduzidos em um site. Usando o elemento `<audio>`, um clipe de áudio pode ser adicionado a uma página. Assim como com o elemento `<img>`, um elemento `<audio>` também precisa de um URL de origem especificado por meio do atributo “src”. Confira um exemplo:



The screenshot shows a code editor window titled "Untitled" with the subtitle "Captain Anonymous". The tab bar indicates the file is an "HTML" file. The code in the editor is as follows:

```
1 <html>
2
3 <head>
4   <title>Exemplo</title>
5 </head>
6
7 <body>
8   <p>Clique para reproduzir:</p>
9   <audio controls>
10    <source src="https://2u039f-a.akamaihd.net/downloads/ringtones/files/mp3/kal-ho-na-ho-18413-50654.mp3"
11      type="audio/mp3">
12
13 </body>
14
15 </html>
```

To the right of the code editor, there is a preview area with a play button, a progress bar showing "0:03 / 0:27", and other media controls.

Figura 4 - Elemento de áudio em HTML

Fonte: do Autor (2022)

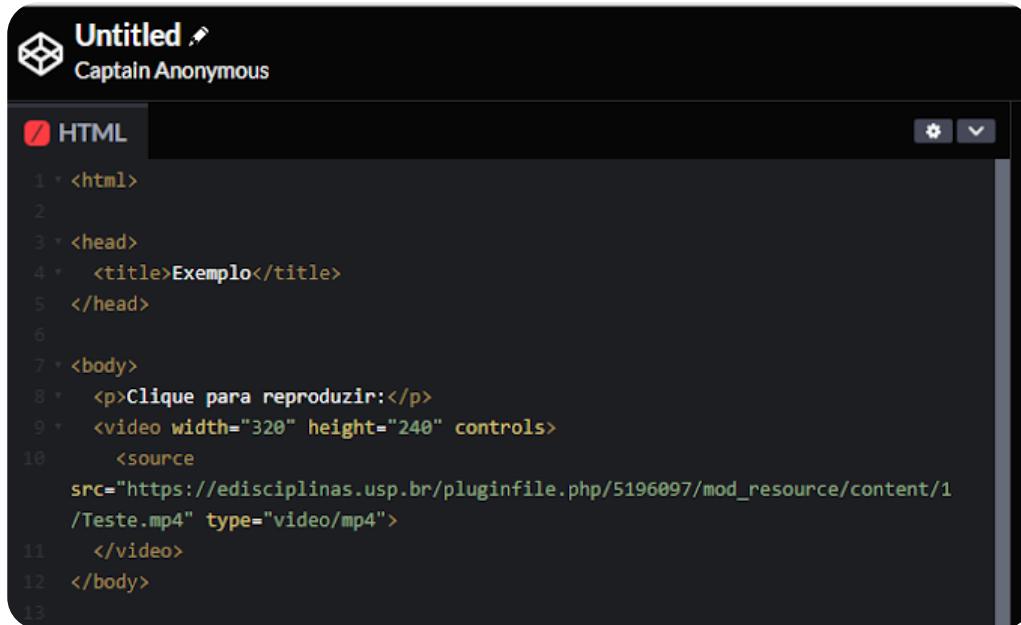


### Atenção .....

Cada um dos possíveis atributos do elemento `<audio>` tem uma finalidade:

- O atributo “src” é destinado a indicar o caminho para o arquivo de som a incorporar na página.
- O atributo “controls” é destinado a fazer com que o navegador renderize os controles nativos para o usuário interagir com o som.
- O atributo “autoplay” executa o som automaticamente.
- O atributo “loop” repete a execução do som ao seu final.
- O atributo “muted” desativa o som.

A adição de vídeos HTML5 é muito semelhante à adição de áudio. Neste caso, no entanto, usamos o elemento <video> no lugar do elemento <audio>. Entenda com o exemplo de código a seguir:



The screenshot shows a code editor window titled "Untitled" with the subtitle "Captain Anonymous". The tab bar indicates the file is an HTML document. The code in the editor is as follows:

```
1 <html>
2
3 <head>
4   <title>Exemplo</title>
5 </head>
6
7 <body>
8   <p>Clique para reproduzir:</p>
9   <video width="320" height="240" controls>
10    <source
11      src="https://edisciplinas.usp.br/pluginfile.php/5196097/mod_resource/content/1
12      /Teste.mp4" type="video/mp4">
13    </video>
14 </body>
15
```

Clique para reproduzir:



Figura 5 - Elemento de vídeo em HTML

Fonte: do Autor (2022)

## Estrutura do DOM

DOM significa *Document Object Model*. É uma interface de programação que nos permite criar, alterar ou remover elementos do documento HTML. Também podemos adicionar eventos a esses elementos para tornar nossa página mais dinâmica. O DOM visualiza um documento HTML como uma árvore de nós. Um nó representa um elemento HTML (WHATWG, 2022).

Imagine que, cada página da web é um documento e cada elemento (seja texto, imagem ou tabela) dentro dessa página da web é armazenado nesse documento como um objeto em formato de árvore, que pode ser manipulado posteriormente com uma linguagem de script, como JavaScript. Agora, quando uma página da web é carregada, o navegador cria tal documento toda vez que é chamado.

O DOM representa o documento como nós e objetos para que os programas possam alterar a estrutura, o estilo e o conteúdo do documento. Dessa forma, as linguagens de programação se conectam à página da web.

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Estrutura de árvore DOM</title>
</head>

<body>
    <h1>Estrutura de árvore DOM</h1>
    <h2>Aprendendo sobre DOM</h2>
</body>

</html>
```

Vamos dar uma olhada neste código HTML para entender melhor a estrutura de árvore do DOM.

Nosso documento é chamado de nó raiz e contém um nó filho que é o elemento <html>. O elemento <html> contém dois filhos que são os elementos <head> e <body>. Ambos os elementos <head> e <body> têm seus próprios filhos. Aqui está outra maneira de visualizar essa árvore de nós.

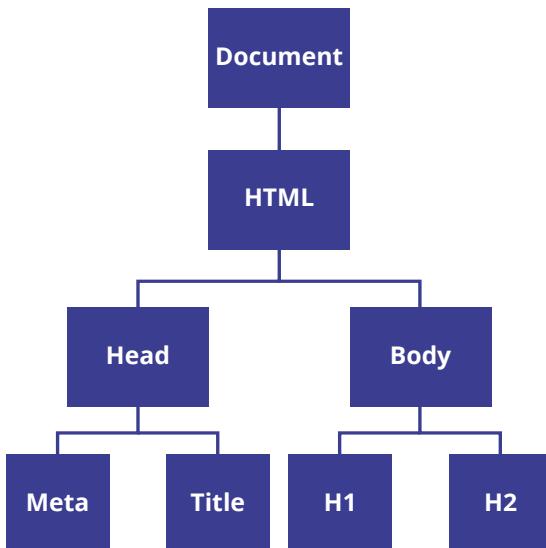


Figura 6 - Estrutura da árvore DOM

Fonte: do Autor (2022)



#### Dica .....

Perceba que o DOM não é uma linguagem de programação, mas sem ele, a linguagem JavaScript não seria capaz de acessar nenhum dos elementos de páginas web em documentos HTML, documentos XML ou suas partes componentes. Com o DOM, o JavaScript obtém todo o poder necessário para criar HTML dinâmico, como alterar os elementos HTML, atributos, estilos CSS, elementos e eventos na página.



## Formulários

Os formulários são uma das partes mais importantes da web. Sem eles, não haveria uma maneira fácil de coletar dados, pesquisar recursos ou se inscrever para receber informações valiosas. Você pode incorporar formulários em sites com o elemento <form> do HTML. Dentro do elemento de formulário, várias entradas são aninhadas. Essas entradas também são conhecidas como controles de formulário.

Nesta seção, exploraremos o elemento de formulário HTML, os vários tipos de entrada necessários e como criar um botão de envio com o qual os dados são enviados. No final, você saberá como os formulários funcionam e poderá fazê-los com segurança. Confira, a seguir, a sintaxe básica de um formulário HTML:

```
<form action="www.site.com.br" method="post">  
    <!--Entrada de qualquer tipo e áreas de texto entram  
        aqui...-->  
</form>
```

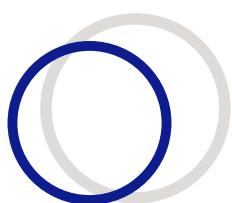
Entenda alguns atributos presentes na tag `<form>` no quadro a seguir.

Atributo	Propósito	Valor
action	Menções para onde os dados do formulário devem ser enviados após o envio.	URL
autocomplete	Menciona se o formulário tem recurso de preenchimento automático ou não.	on off
method	Especifica o método usado para enviar os dados do formulário.	get post
name	Nome do formulário.	texto

Quadro 4 - Atributos na tag `<form>`

Fonte: do Autor (2022)

Você usa a tag `<input>` para criar vários controles de formulário em HTML. Desta forma, a tag `<input>` especifica uma área para capturar a entrada do usuário. É um elemento embutido e recebe atributos como “type”, “name”. Cada um deles assume valores específicos. Confira alguns atributos utilizados pela tag `<input>`:



Atributo	Propósito	Valor
alt	Menciona um texto alternativo para aparecer se a imagem estiver faltando.	texto
autofocus	Menciona se o campo de entrada deve ter foco quando o formulário é carregado.	autofocus
name	Menciona o nome do campo de entrada.	texto
required	Menciona se um campo de entrada é obrigatório.	required
size	Menciona o comprimento do caractere.	número
type	Menciona o tipo de campo de entrada.	button, checkbox, image, password, radio, text, time (entre outros)
value	Menciona o valor de uma área de entrada.	texto
placeholder	Ajuda o usuário a entender a finalidade do campo de entrada antes de digitar qualquer coisa.	texto
hidden	Um atributo oculto em uma tag <input> oculta o elemento de entrada. Embora o elemento de entrada não esteja visível, sua posição na página é mantida.	hidden

Quadro 5 - Atributos na tag <input>

Fonte: do Autor (2022)

Os atributos “disabled” e “readonly” têm efeitos semelhantes. Na verdade, eles têm a mesma aparência da perspectiva do usuário. Em ambos os casos, o que o usuário vê é um campo levemente “esmaecido”, que não pode ser clicado ou editado. Compreenda com o exemplo:

```

● ● ●
<form>
  <label for="txtNome">Nome</label><br>
  <input name="txtNome" disabled>
  <br><br>
  <label for="txtCargo">Cargo</label><br>
  <input name="txtCargo" readonly>
</form>

```

Se um campo for “disabled”, o valor do campo não será enviado ao servidor quando o formulário for enviado. Se um campo for “readonly”, o valor será enviado ao servidor.

Existem vários tipos de entrada diferentes, e vamos analisar alguns deles no quadro apresentado na sequência.

Tipos de entradas	Propósito	Exemplo
text	Esse tipo de entrada assume um valor de “texto”, então cria uma única linha de entrada em formato textual.	<input type="text" placeholder="Digite seu nome" />
password	Este tipo de entrada cria um campo de senha. É automaticamente invisível para o usuário, a menos que seja manipulado por JavaScript.	<input type="password" placeholder="Digite sua senha" />
email	Esta entrada define um campo para inserir um endereço de e-mail. Neste caso, este campo valida a entrada usando o formato de e-mail padrão.	<input type="email" placeholder="Digite seu e-mail" />
number	Este tipo de entrada permite que o usuário insira apenas números.	<input type="number" placeholder="Digite um número" />
radio	Este tipo de campo permite que os usuários escolham uma entre várias opções.	<input type="radio" />
checkbox	Este tipo de entrada permite que os usuários possam escolher várias opções ao mesmo tempo.	<input type="checkbox" />
file	Isso define um campo para envio de arquivo. Quando um usuário clicar nele, será solicitada a inserção de um tipo de arquivo desejado.	<input type="file" />
color	Com este campo, o usuário pode selecionar uma cor. Permite inserir um valor de cor simples (que está em notação de cor hexadecimal).	<input type="color" />

search	A entrada com o tipo de pesquisa define um campo de texto exatamente como um tipo de entrada de texto. Mas, desta vez, tem o único propósito de buscar informações. É diferente do tipo de texto, pois um botão de cancelamento aparece quando o usuário começa a digitar.	<code>&lt;input type="search" /&gt;</code>
url	Quando o atributo de uma tag de entrada é definido como URL, ele exibe um campo no qual os usuários podem inserir apenas um URL válido.	<code>&lt;input type="url" /&gt;</code>
tel	Um tipo de entrada de telefone permite coletar números de telefone dos usuários.	<code>&lt;input type="tel" /&gt;</code>
time	O elemento de entrada com um atributo de tipo de tempo permite definir um horário do dia, embora não apresente informações de fuso horário.	<code>&lt;input type="time" /&gt;</code>
date	Este campo é um controle para inserir uma data.	<code>&lt;input type="date" /&gt;</code>
datetime	Este campo é utilizado para inserção de data e hora usando o formato UTC ( <i>Universal Time Coordinated</i> ), ou seja, utilizando um fuso horário.	<code>&lt;input type="datetime" /&gt;</code>
datetime-local	Isso funciona como um tipo de entrada de data, mas também permite que o usuário escolha uma data com um horário específico. Assim, data e hora são informadas de acordo com a hora local, sem informações de fuso horário.	<code>&lt;input type="datetime-local" /&gt;</code>
week	O tipo de entrada de semana permite que um usuário selecione uma semana específica e o ano.	<code>&lt;input type="week" /&gt;</code>



month	A entrada com o tipo de mês preenche meses para o usuário escolher quando clicado.	<pre>&lt;input type="month" /&gt;</pre>
range	Um controle deslizante para escolher um número entre determinado intervalo.	<pre>&lt;input type="range"&gt;</pre>
textarea	O campo do tipo "textarea" permite preencher várias linhas de texto. Ele recebe seus próprios atributos, como "cols" para o número de colunas e "rows" para o número de linhas.	<pre>&lt;textarea cols="20" rows="20"&gt;&lt;/textarea&gt;</pre>
select	Este campo permite que uma caixa de seleção e um botão de opção sejam agrupados, sendo um elemento "select" e outro elemento "option". A tag <option> é usada para definir as opções de uma lista <select>. Por padrão, o usuário só pode escolher uma das opções.	<pre>&lt;select&gt;     &lt;option value=""&gt;-     Selecione-&lt;/option&gt;     &lt;option value="HTML"&gt;HTML&lt;/option&gt;     &lt;option value="CSS"&gt;CSS&lt;/option&gt;     &lt;option value="JavaScript"&gt;JavaScript&lt;/option&gt; &lt;/select&gt;</pre>
submit	Este campo é usado para adicionar um botão de envio aos formulários. Quando um usuário clicar nele, o formulário será enviado automaticamente. Além disso, é possível definir um atributo "value" para determinar o texto que aparece dentro do botão.	<pre>&lt;input type="submit" value="Enviar" /&gt;</pre>
button	Uma entrada com um tipo definido como "button" cria um botão, entretanto ele deve ser manipulado pelo tipo de evento "onClick" do JavaScript. Desta forma, estes botões não enviarão um formulário, logo eles não fazem nada por padrão.	<pre>&lt;input type="button" value="Enviar" /&gt;</pre>

Quadro 6 - Tipos de entrada

Fonte: do Autor (2022)

Além de todos estes recursos apresentados, atribuir rótulos aos controles de formulário é importante. Quando eles estão conectados corretamente ao campo de entrada por meio de seu atributo “for” e do atributo “id” de entrada, é mais fácil para o usuário usar, pois basta clicar no próprio rótulo para acessar a entrada. Confira um exemplo:

```
<form action="www.site.com.br" method="post">
    <label for="txtNome">Nome</label>
    <input type="text" id="txtNome" /> <br />
    <label for="chkTermos">Eu aceito os termos!</label>
    <input type="checkbox" id="chkTermos" />
</form>
```

Um grupo de rádio é definido dando a cada um dos botões de rádio no grupo o mesmo “name”. Depois que um grupo de rádio é estabelecido, a seleção de qualquer botão de opção nesse grupo desmarca automaticamente qualquer botão de opção atualmente selecionado no mesmo grupo. Entenda melhor a partir de um exemplo:

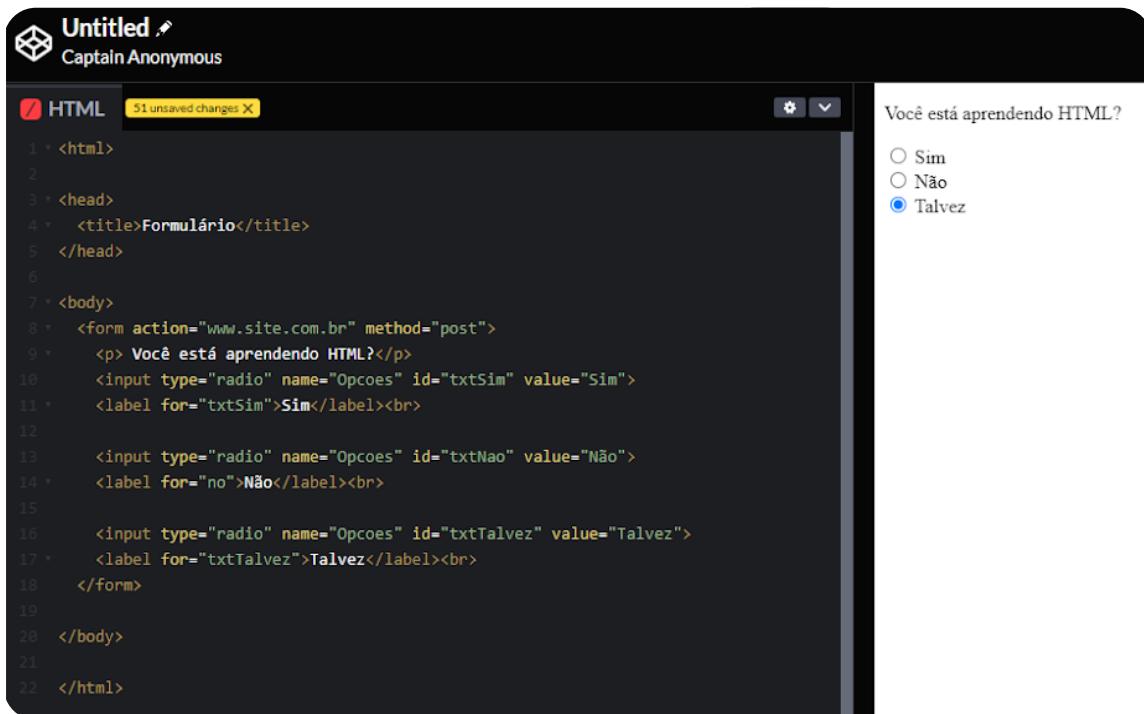


Figura 7 - Exemplo de um campo do tipo radio

Fonte: do Autor (2022)



Quando um usuário preenche um formulário e o envia com o botão de envio, os dados nos controles de formulário são enviados ao servidor por meio de métodos GET e POST de solicitação HTTP. Então, como o servidor é indicado? O elemento <form> recebe um atributo “action”, que deve ter seu valor especificado para o URL do servidor. Ele também recebe um atributo de método, onde o método HTTP usado para transmitir os valores ao servidor é especificado.

Este método pode ser GET ou POST. Com o método GET, os valores inseridos pelo usuário ficam visíveis no URL quando os dados são enviados. Mas com POST, os valores são enviados em cabeçalhos HTTP, portanto, esses valores não são visíveis no URL. Se um atributo de método não for usado no formulário, assume-se automaticamente que o usuário deseja usar o método GET, pois é o padrão (WHATWG, 2022).



### Reflita

Então, quando você deve usar o método GET ou POST? Use o método GET para enviar dados não confidenciais ou recuperar dados de um servidor (por exemplo, durante pesquisas). Use a solicitação POST ao enviar arquivos ou dados confidenciais.

Vamos aplicar o que aprendemos sobre formulários e usá-lo para fazer um formulário de contato simples. Você também confere alguns novos conceitos à medida que formos completando tudo. Na sequência, é apresentado o formulário:



Captain Anonymous

HTML

```

1 <html>
2
3 <head>
4   <title>Formulário</title>
5 </head>
6
7 <body>
8   <form action="www.site.com.br" method="post">
9     <fieldset>
10       <legend>Formulário de Contato</legend>
11       <div class="campos">
12         <label for="txtNome">Nome</label>
13         <input type="text" id="txtNome" placeholder="Digite seu nome" required />
14       </div>
15
16       <div class="campos">
17         <label for="txtEmail">E-mail</label>
18         <input type="email" id="txtEmail" placeholder="Digite seu e-mail" required />
19       </div>
20
21       <div class="campos">
22         <label for="txtMensagem">Mensagem</label>
23         <textarea id="txtMensagem" cols="30" rows="10" placeholder="Digite sua mensagem" required></textarea>
24       </div>
25       <input type="submit" value="Enviar" />
26     </fieldset>
27   </form>
28
29 </body>
30
31 </html>

```

Figura 8 - Exemplo de um formulário

Fonte: do Autor (2022)

Vamos analisar o código apresentado. Primeiro, um elemento `<form>` está envolvendo todos os outros elementos. Ele tem uma ação definida para “`www.site.com.br`”, um servidor fictício onde os dados do formulário serão recebidos. Após o elemento de formulário, todos os outros elementos também são cercados por um elemento `<fieldset>` com uma tag `<legend>` logo abaixo dele. Usamos o elemento `<fieldset>` para agrupar entradas relacionadas e a tag `<legend>` contém uma legenda que transmite sobre o que é o formulário.

As entradas do tipo “`text`”, “`email`”, e “`textarea`” estão todas envolvidas com uma `<div>` com uma classe de controle de formulário. Portanto, eles se comportam como um elemento de bloco, para facilitar o estilo com CSS, quando necessário.

Eles também são validados com o atributo “required”, portanto, o formulário não é enviado quando esses campos estão vazios ou quando o usuário não digita os valores no formato apropriado.

Neste estudo você conheceu os recursos de listas e tabelas. Acompanhou como inserir imagem, áudio e vídeo em páginas web. Posteriormente, entendeu a importância e o funcionamento do DOM (*Document Object Model*). Por fim, conheceu diversos recursos relacionados aos formulários. Para cada uma das tags, apresentamos as tags comumente utilizadas juntamente com seus respectivos atributos, além disso, disponibilizamos trechos de código e resultado esperado para ilustrar suas aplicações.



Este estudo foi desenvolvido para te ajudar a entender como os formulários e demais recursos do HTML funcionam. Agora você tem o conhecimento necessário para integrar formulários em seus sites para iniciar a coleta de dados.

Esperamos que este conteúdo tenha fornecido a você uma compreensão do que são exatamente os principais recursos do HTML. Continue seus estudos!

## REFERÊNCIAS

WHATWG. HTML Living Standard. Disponível em: <https://html.spec.whatwg.org>. Acesso em: 30 jun. 2022.



