

# Autoria Web

## Aula 03 - HTML: Figuras, Tabelas, Listas e Formulários

# Apresentação

---

Até agora você aprendeu como especificar textos e links em um documento HTML. No entanto, dificilmente um site é composto apenas por esses elementos. Figuras, tabelas, listas e formulários são elementos que tornam os sites muito mais interessantes e fáceis de ler. Nesta aula, estudaremos como inserir esses elementos em documentos HTML.

Caso tenha alguma dúvida sobre a estruturação de um documento HTML, cabeçalho, títulos de seção, parágrafos e links, você deve retornar à **Aula 2** desta disciplina. Lembre-se de que você deve ter à sua disposição os arquivos referentes ao site que você começou a desenvolver na aula passada.

Bons estudos!



## Vídeo 01 - Apresentação

## Objetivos

Ao final desta aula, você deverá ser capaz de:

- Identificar as tags que definem figuras, listas, tabelas, formulários e frames em documentos HTML.
- Criar documentos com esses elementos.

# Figuras

---

A inserção de imagens em páginas HTML é feita na seção de corpo da página (<body>...</body>), através do elemento <img/> ou <object>.

## O elemento <img/>

A tag <img/> pode possuir basicamente 5 atributos: src, alt, longdesc, width e height. A seguir, você conhecerá cada um deles:

**1) src** – Indica o caminho do arquivo referente à figura, incluindo o seu formato (pode ser jpg, gif, png, dentre outros tipos). É interessante organizar as imagens usadas em uma página numa única pasta, para facilitar a manutenção dos arquivos. A seguir, é mostrado, de forma genérica, um exemplo da tag <img/> com o atributo src:

```
1 
```

**2) alt** – Fornece uma descrição curta sobre a imagem quando o cursor move-se sobre a mesma e, por algum motivo, o navegador não consegue mostrá-la. No exemplo que segue, ao posicionar o mouse sobre a figura, o texto "Texto Descritivo" será exibido.

```
1 
```

**3) longdesc** – Especifica um link para uma descrição extensa da imagem. Essa descrição deve complementar a descrição provida pelo atributo alt. No exemplo a seguir, a imagem contém um link para o documento definido no atributo longdesc.

```
1 
```

**4) Width/Height** – Especifica a largura e a altura da imagem, respectivamente. Os valores podem ser em pixels ou proporcionais ao tamanho que a figura ocupará na página.

```
1 
```

Os atributos align, border e vspace/hspace, que seguem, responsáveis pela disposição da figura na página, também podem ser utilizados na tag <img/>, mas eles são **DEPRECATED** (Como foi visto na Aula 2 deste curso, **DEPRECATED** é o termo usado para elementos e atributos que estão em desuso, ou seja, estão sendo substituídos por outros em versões mais recentes dessas linguagens.).

- **Align** – Especifica o tipo de alinhamento entre a imagem e o texto da página. “center”, “left”, “right” e “top” são alguns dos possíveis valores para esse atributo. Eles posicionam a figura de maneira centralizada, à esquerda, à direita ou acima do texto, respectivamente.

```
1 
```

- **Border** – Determina a espessura da borda da imagem. No caso de não haver borda, esse atributo não precisará ser citado.

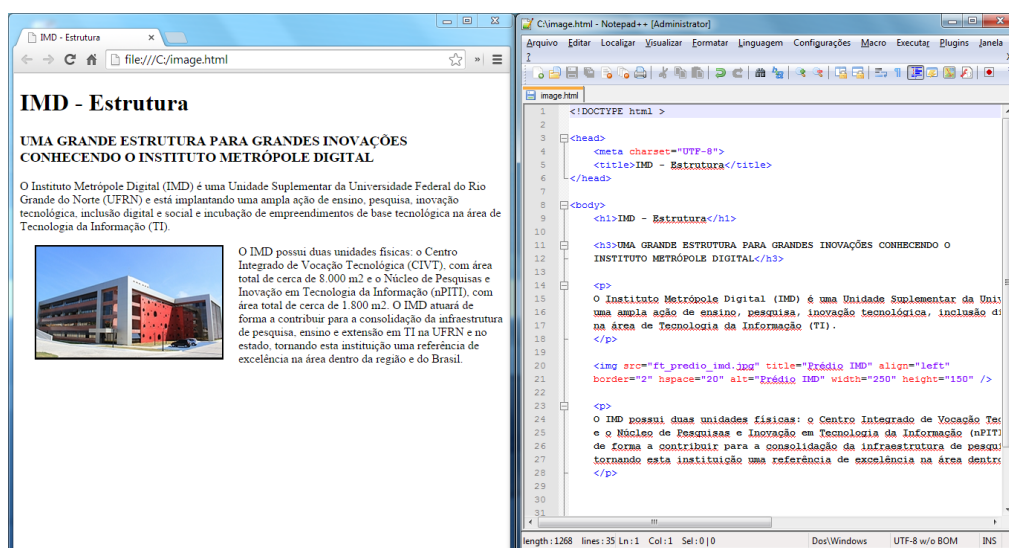
```
1 
```

- **Vspace/Hspace** – Especifica o espaço vertical e horizontal, respectivamente, que pode aparecer entre a imagem e o texto mais próximo ou até mesmo para outra imagem na linha corrente. É dado em pixels.

```
1 
```

A **Figura 1** (lado direito) mostra um código HTML que insere uma imagem na página, mostrada no lado esquerdo da mesma figura.

**Figura 01** - Exemplo do uso do elemento <img> e seus atributos



Os atributos: align, border, hspace, longdesc, and vspace não são mais suportados no HTML5.

## O elemento <object>

Uma imagem pode ser inserida em um documento HTML através do elemento object. Essa tag é mais genérica e pode inserir figuras, vídeos e [applets](#) (Applet é um software aplicativo que é executado no contexto de outro programa (como por exemplo, um web browser). Uma applet geralmente executa funções bem específicas (Wikipédia, 2010. Extraído da internet).) em um documento HTML. Para inserir figuras, os atributos necessários são **data** e **type**. O atributo data fornece a URI da imagem e o atributo type pode ter o valor image/jpg indicando que se trata de uma figura com formato jpg. Os exemplos a seguir com as tags **<img/>** e **<object>** são equivalentes.

```
1 
```

Ou

```
1 <object data="URI" type="image/jpg" width="100" height="100" >descricao da imagem </object>
```

Observe que enquanto a tag <img/> é vazia, a tag <object> não é.

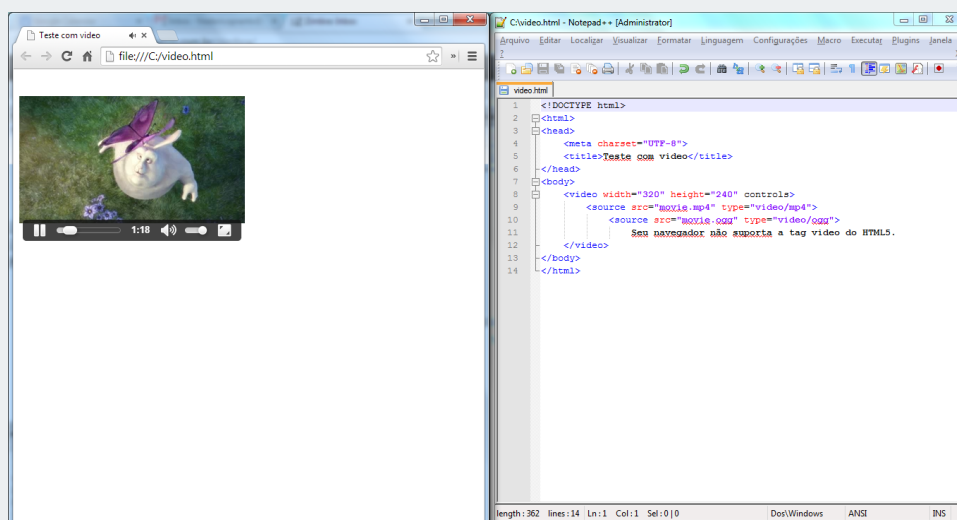


**Vídeo 02** - Imagem

A tag <video> do HTML5 especifica um padrão para exibição de vídeos em páginas web. Antes do HTML5 não havia um padrão e era necessário instalar plug-ins de terceiros para a execução de vídeos. A seguir vamos mostrar um exemplo de como exibir um vídeo em HTML5.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Teste com video</title>
6   </head>
7   <body>
8     <video width="320" height="240" controls>
9       <source src="movie.mp4" type="video/mp4">
10      <source src="movie.ogg" type="video/ogg">
11      Seu navegador não suporta a tag video do HTML5.
12    </video>
13  </body>
14 </html>
```

**Figura 02**



O vídeo do exemplo está licenciado sobre o [Creative Commons](http://creativecommons.org/licenses/by/3.0/) (As licenças Creative Commons são um conjunto de licenças de direito autoral que garantem alguns direitos sobre a obra, como o direito de distribuir obras sem modificações, a custo zero.) Attribution 3.0 e pode ser baixado no seguinte endereço <http://www.bigbuckbunny.org/>

# Player de Áudio HTML5

A música esta disponível [aqui](#).

## Atividade 01

---

1. Insira imagens no documento sobre você como, por exemplo, uma foto sua, de seus familiares, capas de livros ou pôsteres de filmes que você gosta. Organize-os de diferentes maneiras, usando todos os atributos mencionados anteriormente.
2. Insira links em algumas das imagens incluídas na questão anterior. Lembre-se de que para adicionar um link é necessário usar a tag `<a>` e seus atributos, explicados na Aula anterior. Genericamente, você define um link em uma imagem da seguinte maneira:

1	<code>&lt;a href="URI"&gt;&lt;object ...&gt;&lt;/object&gt;&lt;/a&gt;</code>
---	--

Guarde esse documento, pois você ainda irá utilizá-lo nas próximas atividades.

## Listas

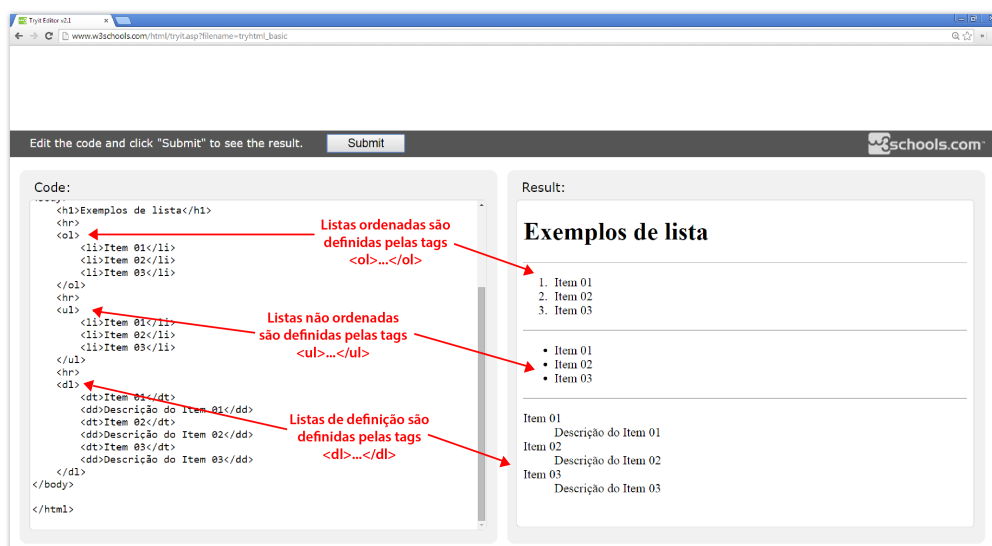
---

Existem três tipos de listas: **ordenadas**, **não ordenadas** e **de definição**. Cada tipo de lista é delimitada por uma tag diferente:

- **`<ol>...</ol>`** – para listas ordenadas (*ordered list*). Essas listas são utilizadas quando se quer numerar algo;
- **`<ul>...</ul>`** – para listas não ordenadas (*unordered list*). Essas listas são utilizadas quando se quer apenas colocar um marcador em seus itens, mas sem numeração;
- **`<dl>...</dl>`** – para listas de definição (*definition list*). Essas listas são usadas quando, além de topificar, ainda se deseja especificar uma definição para cada tópico.

A tag `<li>...</li>` é usada para marcar cada item das listas ordenadas e não ordenadas. Na lista de definição, os itens são marcados pela tag `<dt>...</dt>`, enquanto as definições são marcadas pela tag `<dd>...</dd>`. A Figura 3 ilustra um exemplo de cada tipo de lista, veja com atenção como cada uma delas é definida.

**Figura 03** - Exemplos de listas



### Vídeo 03 - Criando Listas

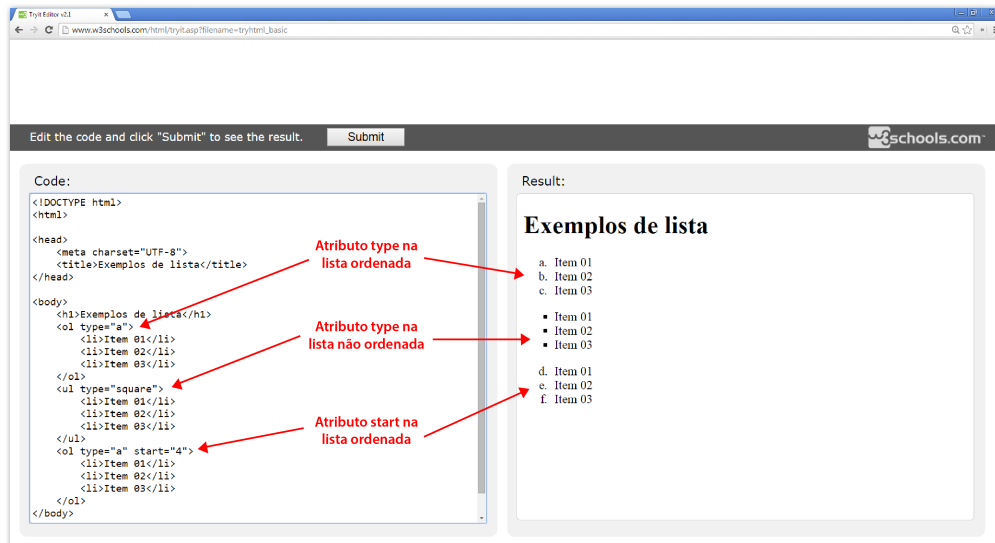
Há alguns atributos que podem ser utilizados para configurar a aparência das listas, tais como `start` e `type`, mas lembre-se que eles são DEPRACATED:

- **start** – indica o número do primeiro item de uma lista ordenada;
- **type** – indica o tipo de numeração (arábico, romano...) em listas ordenadas ou o tipo de símbolo ("square", "circle" ou "disc") usado para demarcar cada item de uma lista não ordenada.

A seguir, você verá um exemplo simples desses atributos (Figura 4). Mas você não precisa memorizá-los, pois logo (a partir da quarta aula) veremos como definir o mesmo efeito usando CSS.



**Figura 04** - Exemplos de listas com atributos type e start



Os atributos **start** e **type** da tag `<ol>` foram descontinuados no HTML4 mas são suportados no HTML5, além disso foi adicionado o atributo **reversed** para representar a lista na ordem inversa.

Agora, volte àquele documento criado para a **Atividade 1** e resolva a próxima atividade.

## Atividade 02

1. Troque os parágrafos simples do documento sobre você pelo tipo de lista mais adequado, fazendo com que seu documento possua pelo menos uma lista de cada tipo. Caso você ainda não tenha inserido conteúdo suficiente para isso, aproveite para fazê-lo agora.

## Tabelas

Tabelas são definidas com a tag `<table>...</table>`. Elas são divididas em linhas, delimitadas pela tag `<tr>...</tr>`, e colunas, delimitadas pela tag `<td>...</td>`, nas quais se determina o conteúdo das células. Esse conteúdo pode ser texto, imagens,

listas, formulários ou outros elementos.

O elemento table pode conter os atributos:

- **border**, que indica qual o tipo de borda da tabela (se não for definido, nenhuma borda aparecerá);
- **width**, que indica a largura da tabela (ele pode ser dado em percentual);
- **summary**, que fornece uma descrição da tabela, útil em termos de acessibilidade (na Aula 10 você aprenderá sobre acessibilidade).

Veja em detalhes 2 exemplos de tabela na **Figura 5**.

**Figura 05** - Exemplos de tabela

The screenshot shows a web browser window with a code editor on the left and a rendered table on the right. The code editor contains two HTML table examples. The first example is a simple 2x2 table with a border attribute. The second example is a 2x2 table with a width attribute and header rows. Red arrows point from the code to the rendered table, explaining the attributes and tags used.

**Code:**

```
<table border="1px">
  <tr>
    <td>Linha 1, Coluna 1</td>
    <td>Linha 1, Coluna 2</td>
  </tr>
  <tr>
    <td>Linha 2, Coluna 1</td>
    <td>Linha 2, Coluna 2</td>
  </tr>
</table>

<table width="100%" border="1px">
  <tr>
    <th>Titulo 1</th>
    <th>Titulo 2</th>
  </tr>
  <tr>
    <td>Linha 1, Coluna 1</td>
    <td>Linha 1, Coluna 2</td>
  </tr>
  <tr>
    <td>Linha 2, Coluna 1</td>
    <td>Linha 2, Coluna 2</td>
  </tr>
</table>
```

**Result:**

**Exemplos de tabela**

Linha 1, Coluna 1	Linha 1, Coluna 2
Linha 2, Coluna 1	Linha 2, Coluna 2

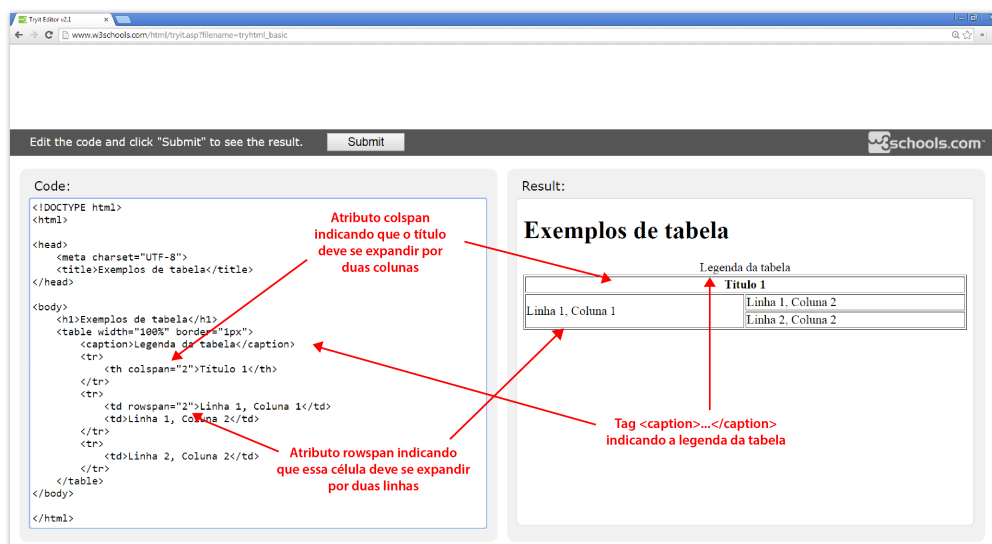
Titulo 1	Titulo 2
Linha 1, Coluna 1	Linha 1, Coluna 2
Linha 2, Coluna 1	Linha 2, Coluna 2

Annotations:

- Tag <table>...</table> com o atributor border se indica as bordas da tabela
- Tags <tr>...</tr> delimitam as linhas e tags <td>...</td> as colunas
- Atributo width indica a largura da tabela
- Tag <th>...</th> define títulos para as colunas da tabela

Além das tags **<td>**, o elemento **<tr>** pode conter a tag **<th>**, que indica um título para a linha/coluna. Esse título aparecerá em negrito e centralizado. No exemplo ilustrado na **Figura 5**, a segunda tabela possui títulos para as colunas. Dentre os possíveis atributos das tags **<tr>** e **<td>** existem os atributos **colspan** e **rowspan**, que servem para indicar a quantidade de células (linha ou coluna) que serão ocupadas por um certo conteúdo. Por exemplo, a **Figura 6** ilustra uma tabela com um título para as duas colunas.

**Figura 06** - Exemplo de tabela com o elemento caption e os atributos colspan e rowspan



Os atributos **border**, **width** e **summary** não são mais suportados no HTML5.

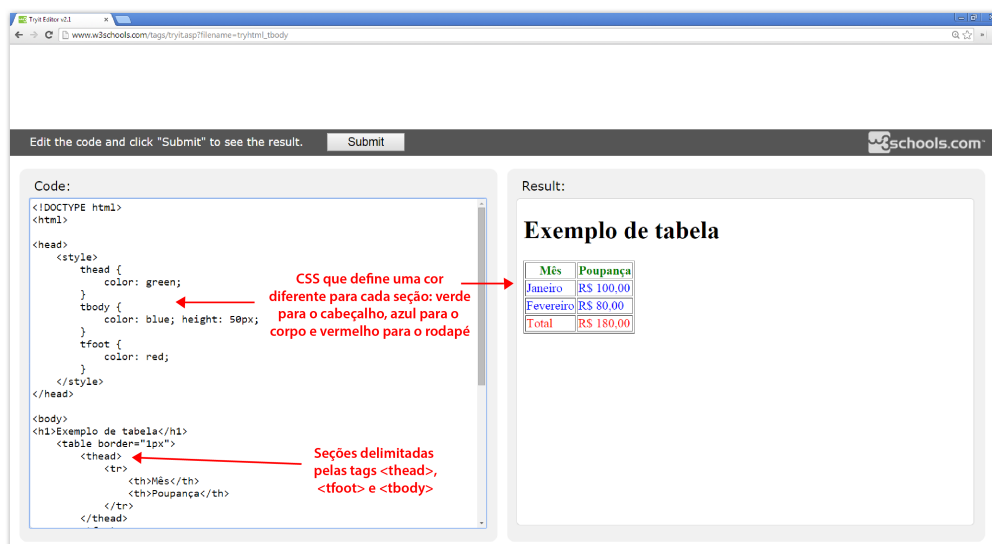
## Elemento caption

Observando a **Figura 6**, você deve ter visto a tag **<caption>**. Ela define uma legenda para a tabela. Essa tag deve ser inserida como a primeira subtag de **<table>** e não deve aparecer mais de uma vez. Usualmente, essa legenda aparecerá centralizada acima da tabela.

## Os elementos thead, tbody e tfoot

As tags **<thead>**, **<tbody>** e **<tfoot>** são usadas para agrupar informações sobre o cabeçalho, o conteúdo e o rodapé da tabela, respectivamente. Esses elementos não afetam a aparência da tabela, mas podem ser selecionados pelo CSS de forma que se possa associar uma aparência para cada uma dessas seções. Em versões anteriores do HTML só era permitido a tag **<tfoot>** ocorrer antes de qualquer tag **<tbody>**, no entanto, agora a tag **<tfoot>** também pode ocorrer depois de qualquer tag **<tbody>**, ou seja, ambas as formas são permitidas.

**Figura 07** - Exemplo de tabela com os elementos thead, tfoot e tbody



**Fonte:** dados de: [http://www.w3schools.com/tags/tryit.asp?filename=tryhtml\\_tbody](http://www.w3schools.com/tags/tryit.asp?filename=tryhtml_tbody). Acesso em: 10 fev. 2012.



## Vídeo 04 - Tabelas

## Atividade 03

1. Inclua a tabela seguinte no documento sobre você e troque o conteúdo por informações suas.

coisas que quero fazer.

### O quê

### Quando

Aprender a dançar forró

Antes de entrar na faculdade

Ir à Disney

2013

Pular de asa delta

Quando me formar

O quê	Quando
Comprar um Land Rover	Quando eu estiver ganhando bem
Viajar pela Europa	Antes dos 30 anos
Carnatal	Todos os anos

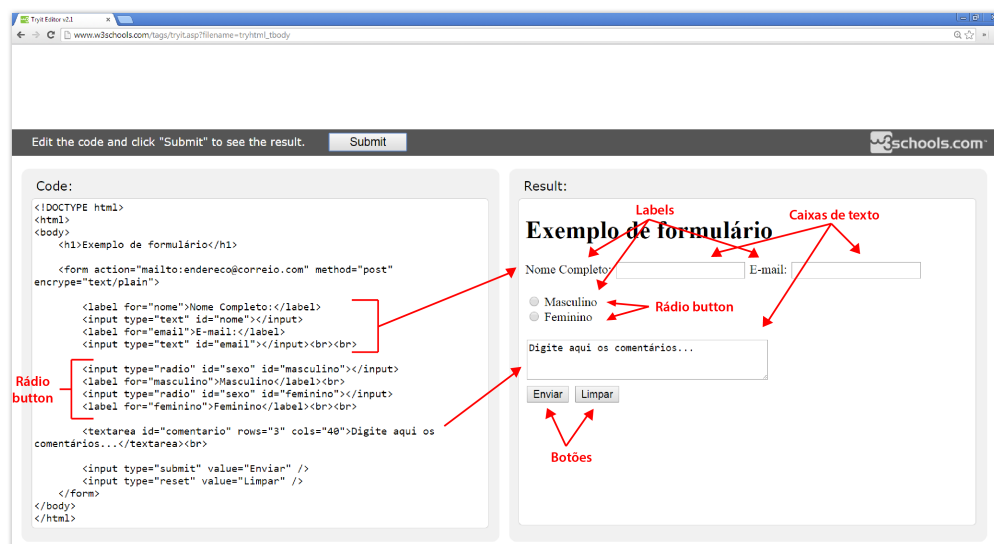
## Formulários

Formulários são o meio pelo qual os visitantes podem trocar informações com o site (o dono ou setor responsável), tais como comprar um produto, preencher uma enquete, solicitar uma informação, inscrever-se em um evento, enviar um comentário para o autor, dentre outros.

Os dados que o usuário insere nos campos de um formulário são enviados ao correio eletrônico do administrador do formulário ou podem ser processados por programas (escritos em uma linguagem de programação tal como Java, PHP, ASP, dentre outras).

Em HTML, formulários são definidos pela tag **<form>...</form>**. Um formulário pode conter elementos de entrada de dados (também chamados de elementos de controle), tais como campos de texto, caixas de checagem, botões, dentre outros, listados a seguir. A **Figura 8** apresenta um exemplo simples de formulário, observe os elementos que foram definidos nele.

**Figura 08 - Exemplo de formulário**



Três dos importantes atributos da tag <form> são:

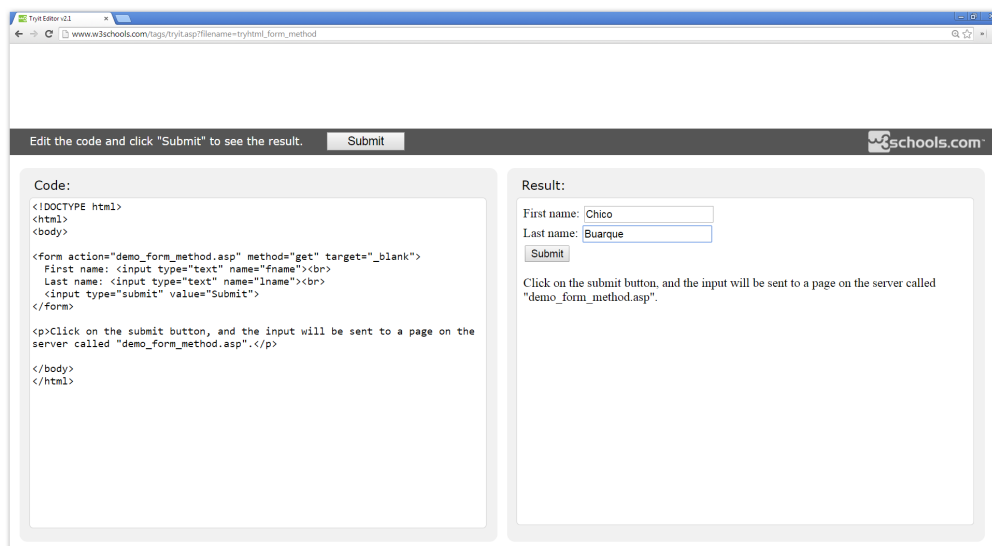
- **action** = "uri" – especifica para onde enviar as informações inseridas no formulário.
- **enctype** – O atributo enctype define como o form-data deve ser codificado quando é submetido para o servidor. Esse atributo com valor "text/plain" é utilizado quando se deseja que os dados do formulário sejam enviados em "texto limpo", sem codificação. Os espaçamentos no texto são convertidos em símbolos de '+'. Assim, conseguimos que o conteúdo do formulário seja enviado como texto plano dentro do e-mail, como no exemplo que segue.

1	<form action="mailto:endereco@correio.com" method="post"
2	enctype="text/plain">...</form>

- **method**= get | post – especifica qual método HTTP será usado para submeter os dados do formulário. Se não for especificado, o valor assumido é "get". O método "get" anexa as informações na URL do formulário, sendo assim, há limitação na quantidade de dados que pode ser anexada (browsers diferentes têm diferentes limitações) e não há qualquer segurança acerca desses dados. Esse método é útil quando se deseja agendar o resultado da submissão dos dados, por exemplo, na lista de favoritos. Já o método post é recomendado quando os dados são privados, tais como senhas.

Na **Figura 9**, ilustramos um formulário utilizando o método get. A **Figura 10** ilustra o que aparece no browser depois que esse formulário é submetido, ou seja, depois que o usuário clica no botão *Submit*. Observe que a URL da **Figura 10** contém os dados inseridos no formulário.

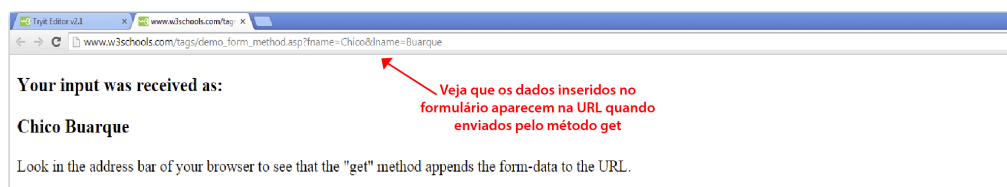
**Figura 09** - Exemplo de formulário utilizando o método "get"



The screenshot shows a web browser window with the URL `www.w3schools.com/tags/tryit.asp?filename=tryhtml_form_method`. The page contains a form with two text input fields labeled "First name:" and "Last name:", and a "Submit" button. The "First name:" field contains the text "Chico" and the "Last name:" field contains the text "Buarque". Below the form, there is a "Result:" section that says "Click on the submit button, and the input will be sent to a page on the server called 'demo\_form\_method.asp'".

Fonte: <[http://www.w3schools.com/tags/att form method.asp](http://www.w3schools.com/tags/att_form_method.asp)>

**Figura 10** - Resultado da submissão do formulário ilustrado na Figura 9



The screenshot shows the browser after the form is submitted. The address bar now shows the URL `www.w3schools.com/tags/demo_form_method.asp?fname=Chico&lname=Buarque`. The page content says "Your input was received as:" followed by "Chico Buarque". A red arrow points to the URL with the text "Veja que os dados inseridos no formulário aparecem na URL quando enviados pelo método get". Below the input, it says "Look in the address bar of your browser to see that the 'get' method appends the form-data to the URL."

Observe que a utilização de formulários está de alguma forma associada a páginas dinâmicas, pois necessita de código (interno ou externo) para tratar os dados enviados pelo formulário, apresentando diferentes resultados de acordo com os dados inseridos. Mesmo que você deseje somente enviar esses dados para um endereço de correio eletrônico, como no exemplo da **Figura 8**, você perceberá que, com o que foi visto até agora em nossas aulas, não é possível.

Isso acontece porque para que o browser consiga enviar os dados de um formulário para um e-mail, é necessário haver um servidor de e-mails instalado onde a página web está hospedada (gravada). Assim, você precisa se preocupar com uma parte importante do desenvolvimento web que é a [hospedagem](#) (Hospedagem

de sites é um serviço que possibilita as pessoas ou empresas manterem suas páginas acessíveis todo o tempo, seja somente pelo protocolo HTTP ou também outros serviços como e-mail, DNS, FTP, dentre outros.).

## Elementos de controle

---

Os elementos de controle de formulários são os elementos que possibilitam a entrada de dados, tais como botões, caixas de texto, check boxes, dentre outros. Vamos começar a explicação pelas caixas de texto.

### Caixa de texto (text input)

Caixas de texto são usadas para que o usuário possa fornecer texto. Elas podem ser definidas com o elemento **<input>**, criando uma única linha de texto, ou com o elemento **<textarea>**, criando múltiplas linhas de entrada de texto. Veja alguns exemplos a seguir.

**1) Texto curto** – na tag **<input>** podem ser especificados dois atributos, **type** e **name**. O atributo **name** serve para identificá-lo em nosso programa de processamento ou no e-mail recebido, enquanto o atributo **type** é importante para indicar o tipo de entrada, pois, como foi visto, a tag **<input>** é usada para criar outros tipos de elemento no formulário. O exemplo a seguir cria uma caixa de entrada de texto com uma única linha e dá o identificador "nome". Veja na **Figura 11** a aparência dessa caixa de entrada de texto.

1	<code>&lt;input type="text" name="nome"&gt;</code>
---	--

Além desses dois atributos, essenciais para o correto funcionamento de nossa etiqueta, existe outra série de atributos que podem ser de utilidade, mas que não são imprescindíveis, tais como:

- **size** – define o tamanho da caixa que aparecerá na tela, em número de caracteres.
- **maxlength** – especifica o tamanho máximo do texto que pode ser inserido em uma caixa de texto. É importante não confundi-lo com o atributo **size**. Enquanto o primeiro define o tamanho aparente da



caixa de texto, maxlength indica o tamanho máximo real do texto que pode ser escrito.

- **value** – define um valor para o campo em questão. Isso pode ajudar o usuário a preencher mais rapidamente o formulário ou a dar alguma ideia sobre a natureza de dados que é solicitada. Esse valor inicial do campo pode ser expresso mediante o atributo value. Veja o resultado com um exemplo simples (**Figura 11**).

```
1 <input type="text" name="nome" value="João"/>
```

**2) Texto oculto** – quando se deseja esconder o texto escrito por meio de asteriscos de forma a fornecer uma certa confiabilidade, o atributo type da tag <input> deve ter valor “password”. Esses campos são ideais para a introdução de dados confidenciais, principalmente senhas. Veja o exemplo a seguir e o efeito na **Figura 11**.

```
1 <input type="password" name="senha"/>
```

**3) Texto longo** – quando se deseja que o usuário escreva várias linhas de texto, deve-se usar a tag <textarea>...</textarea>. Esses campos são práticos quando o conteúdo a enviar não é um nome, telefone ou qualquer outro dado breve, e sim um comentário, mensagem, opinião etc. Na tag <textarea> deve-se indicar, como para o caso visto anteriormente, o atributo name para associar o conteúdo a um nome que será semelhante a uma variável nos programas de processamento e, além disso, pode-se definir as dimensões do campo a partir dos atributos rows e cols. E para escrever um valor inicial para esse campo, deve-se escrevê-lo entre as tags de início e fim de textarea, como no exemplo a seguir.

Veja os exemplos:

- **rows** – define o número de linhas do campo de texto;
- **cols** – define o número de colunas do campo de texto.

```
1 <textarea name="comentário" rows="10" cols="40"></textarea>
```

```
1 <textarea name="comentário" rows="10" cols="40">Escreva seu comentário....</textarea>
```

**Figura 11** - Exemplos de caixas de texto

The diagram illustrates various HTML input types with visual examples and their code. Red arrows connect the code snippets to the corresponding UI elements.

- Caixa de texto curto:** Shows a short text input. The code is:  

```
<label> for="nome">Nome completo: </label>  
<input type="text" name="nome" /><br/><br/>  
<label> for="nome">Nome completo: </label>  
<input type="text" name="nome" value="João" />  
<br/><br/>
```
- Caixa de texto longo:** Shows a long text area. The code is:  

```
<textarea name="comentario" rows="3" cols="40">Digite  
seus comentários aqui...</textarea><br/><br/>
```
- Caixa de texto oculto:** Shows a password input. The code is:  

```
<label> for="senha">Senha: </label>  
<input type="password" name="senha" /><br/><br/>
```
- Caixa de seleção de arquivo :** Shows a file selection input with a 'Procurar...' button. The code is:  

```
<label> for="file">Arquivo: </label>  
<input type="file" name="file" /><br/><br/>
```

A tag **<label>** define um rótulo para os elementos de entrada de dados. Esse rótulo não causa um grande efeito para os usuários, mas melhora a usabilidade porque, se o usuário clicar nesse rótulo, o browser coloca o cursor imediatamente na caixa de controle.

Assim, é aconselhável utilizar essa tag para os elementos de entrada de dados. O atributo "for" da tag **<label>** deve ter o mesmo valor que o atributo "name" do elemento relacionado, como nos exemplos mostrados na **Figura 11**.

## Caixa de seleção de arquivo (file select)

Caixas de seleção de arquivo são controles que permitem que o usuário procure e selecione um arquivo em seu computador para ser submetido em um formulário. Essas caixas também são criadas pelo elemento input, no qual se deve atribuir o valor "file" para o atributo type. Veja um exemplo na **Figura 11**.

1	<code>&lt;input type="file" .../&gt;</code>
---	---

# Botões

Existem 3 tipos de botões:

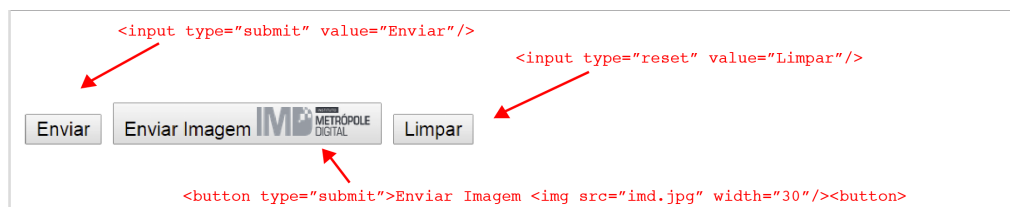
- **Botões submit** – servem para enviar um formulário.
- **Botões reset** – servem para incluir em todos os campos seus valores iniciais ou limpar os campos, caso não tenha sido dado nenhum valor inicial para eles.
- **Botões push** – servem para associar algum comportamento através de scripts.

Os botões são definidos pela tag **<button>** ou **<input>**. Como a tag **<button>** pode não ser vazia, então é possível incluir figuras nos botões, enquanto que com a tag **<input/>** apenas um texto pode ser inserido.

A tag **<button>** pode conter os seguintes atributos: type, para indicar se é do tipo submit, reset ou push; e um name que indicará o nome de controle. O texto que aparece no botão deve ser escrito entre as tags **<button>** e **</button>**. Além de texto, pode-se colocar também figuras entre as tags **<button>** e **</button>**.

A seguir, há três exemplos de botões: o primeiro contém apenas um rótulo; o segundo contém, além do rótulo, uma figura – ambos foram definidos com o tipo submit; enquanto o terceiro botão foi definido com o tipo reset. Observe que para inserir uma figura no botão deve-se usar a tag **<button>** ao invés da tag **<input>**, e é necessário definir um elemento **<img>** dentro da tag **<button>**.

**Figura 12** - Exemplos de botões



# Checkboxes e radio buttons (caixa de seleção e botão de opção)

Checkboxes e radio buttons são elementos que possibilitam ao usuário escolher uma ou várias opções oferecidas pelo formulário. Pode-se selecionar vários checkboxes ao mesmo tempo, mas apenas um radio button (quando um é selecionado, os outros são desmarcados). Esses elementos são definidos pela tag **<input>**.

O atributo **checked** é usado para indicar quais checkboxes e qual radio button devem estar ligados inicialmente, se necessário. A seguir, é definido um exemplo de radio button e um exemplo de checkbox.

```
1 Radio button
2 ...
3 <input type="radio" name="sex" id="Masculino" checked="checked"/>
4 <label for="Masculino">Masculino</label><br/>
5 <input type="radio" name="sex" id="Feminino"/>
6 <label for="Feminino">Feminino</label>
7 ...
```

☒ Masculino  
☐ Feminino

```
1 Checkboxes
2 ...
3 ...
4 <input type="checkbox" name="areap" value="1" id="areap1"/>
5 <label for="areap1">Administração </label>
6 <input type="checkbox" name="areap" value="2" id="areap2"/>
7 <label for="areap2">Educação</label>
8 <input type="checkbox" name="areap" value="3" id="areap3" checked="checked" />
9 <label for="areap3">Informática</label>
```

☐ Administração ☐ Educação ☒ Informática



## Vídeo 05 - Formulário

# Menus

---

Menus são usados para fornecer um conjunto de opções ao usuário. O elemento **<select>**, junto com os elementos **<optgroup>** e **<option>** são usados para definir menus.

O elemento **<select>** cria um menu e cada escolha oferecida pelo menu é representada pelo elemento **<option>**. Um elemento **<select>** deve possuir pelo menos um elemento **<option>**. O elemento **<optgroup>** pode ser usado quando se deseja agrupar as opções. Esse agrupamento é aconselhado quando a lista de opções é extensa.

O elemento **select** pode conter os seguintes atributos:

- **name** – define um nome para o controle.
- **size** – define se um elemento é apresentado como caixa de listas com barra de navegação. Esse atributo especifica o número de linhas que devem ser visíveis ao mesmo tempo. Os navegadores podem apresentar um elemento **<select>** como uma **listbox** ou como um **drop-downmenu**, veja exemplos logo em seguida).
- **multiple** – é um atributo lógico que, se existir, permite a seleção de múltiplas escolhas.

O elemento **<optgroup>** pode conter o atributo **label**, que indica um título para o grupo de opções. O elemento **<option>**, por sua vez, pode conter os seguintes atributos:

- **selected** – se existir, especifica que a opção deve aparecer pré-selecionada quando o menu for carregado.

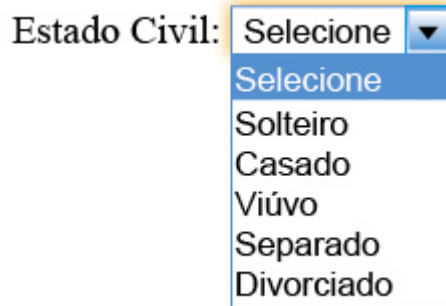
- **value** – especifica o valor a ser enviado para o servidor caso a opção seja selecionada. Se não existir, envia-se o conteúdo da tag como valor.
- **label** – define um nome mais curto para a opção do que aquele do conteúdo da tag. Assim, quando especificado, os browsers exibem esse label ao invés do conteúdo que aparece na tela.

## Exemplo de dropdownmenu

```

1 ...
2 <label for="estado_civil"> Estado Civil: </label>
3 <select name="estadocivil" id="estado_civil">
4 <option label="Selecione" value="-1" selected="selected">Selecione</option>
5 <option label="Solteiro" value="S">Solteiro</option>
6 <option label="Casado" value="C">Casado</option>
7 <option label="Viúvo" value="V">Viúvo</option>
8 <option label="Separado" value="E">Separado</option>
9 <option label="Divorciado" value="D">Divorciado</option> </select>
10 ...

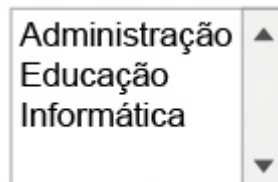
```



```

1 Exemplo de listbox
2 ...
3 ...
4 <select name="area" id="area" multiple="1" size="3">
5 <option label="Administração" value="S">Administração</option>
6 <option label="Educação" value="C">Educação</option>
7 <option label="Informática" value="V">Informática</option></select>
8 ...

```





## Vídeo 06 - Formulário com arquivos e menus

Agora, antes de partir para os exercícios, vamos ler um pouco sobre Javascript? Lembramos que Javascript não faz parte dos assuntos abordados nesta disciplina, mas o encorajamos a pesquisar e estudar o assunto. Que tal começar pelo site da <http://w3schools.com/?!>

### Um parêntese para falar sobre Javascript e validação de formulários

Javascript é uma das mais populares linguagens de script usadas na web e funciona na maioria dos browsers. Essa linguagem foi projetada para adicionar interatividade às páginas HTML. Ela é uma linguagem leve que não precisa ser preliminarmente compilada (é uma linguagem interpretada) e é frequentemente adicionada ao próprio código HTML através do elemento script, visto na segunda aula. Os usos mais comuns da linguagem Javascript são:

- Adicionar texto dinâmico às páginas HTML.
- Reagir a eventos, por exemplo, quando a página termina de carregar ou quando o usuário clica em algum elemento HTML.
- Ler e escrever elementos HTML.
- Validar dados de um formulário antes que ele seja submetido para o servidor. Isso evita que o servidor se ocupe com processamento extra.
- Detectar qual browser está sendo usado e, dependendo de qual for, carregar o documento com a aparência mais apropriada para ele.
- Criar cookies e assim armazenar e recuperar informações no computador visitante.

A seguir, você vai ver um exemplo de Javascript que tem a função de validar formulários. Neste exemplo, utilizamos o mesmo formulário definido na Figura 8, mas observe que a tag <form> possui um atributo a mais, o atributo "onsubmit". Esse atributo é usado para indicar qual script deve ser executado quando o formulário é submetido. Neste caso, o nome do script é validaForm(). Esse script está especificado dentro da tag <script>...</script> na seção de cabeçalho <head>.

Figura 13 - Exemplo de Javascript

```

<html>
<head>
<title>Exemplo de Javascript</title>
<script language="JavaScript">
function validaForm() {
//validar nome
d = document.cadastro;
if (d.nome.value == "") {
alert("O campo " + d.nome.id + " de ser preenchimento!");
d.nome.focus();
return false;
}
//validar email
if (d.email.value == "") {
alert("O campo " + d.email.id + " deve ser preenchido!");
d.email.focus();
return false;
}
//validar email(verificacao de endereco eletrônico)
partel = d.email.value.indexOf("@");
parte2 = d.email.value.indexOf(".");
parte3 = d.email.value.length;
if (!(partel >= 3 && parte2 >= 6 && parte3 >= 9)) {
alert("O campo " + d.email.id + " deve ser conter um endereco eletrônico!");
d.email.focus();
return false;
}
//validar sexo
if (!d.sex[0].checked && !d.sex[1].checked) {
alert("Escolha o sexo!");
return false;
}
return true;
}
</script>
</head>
<body>
<h1>Validação de formulário por javascript</h1>
<form name="cadastro" actio="teste.php" method="post" onsubmit="return validaForm()">
<label for="nome">Nome completo: </label>
<input type="text" id="nome"/><br/>
<label for="email">Email: </label>
<input type="text" id="email"/><br/>
Sexo:
<input type="radio" name="sex" id="Masculino"/>
<label for="Masculino"> Masculino </label>
<input type="radio" name="sex" id="Feminino">
<label for="Feminino"> Feminino </label><br/>
<textarea id="comentario" rows="3" cols="40">Digite seus comentários aqui...</textarea><br>
<input type="submit" value="Enviar"/>
<input type="reset" value="Limpar"/>
</form>
</body>
</html>

```

**JavaScript Section:**

- Nome da função javascript utilizada no atributo onsubmit do elemento form.** (points to `validaForm()`)
- Esta função abre uma janela com a mensagem passada como parâmetro (mensagem entre os parênteses).** (points to `alert()` calls)
- Esta função coloca o cursor no campo indicado(neste caso, no campo "nome").** (points to `d.nome.focus()`)
- O retorno do valor falso faz com que o formulário não seja enviado para o servidor.** (points to `return false;`)
- Verifica se o campo "nome" foi preenchido.** (points to `d.nome.value == ""`)
- Verifica se o campo "email" foi preenchido.** (points to `d.email.value == ""`)
- As variáveis partes e parte2 recebem a posição do "@" e do ".", enquanto variável partes recebe a quantidade de caracteres do email.** (points to email validation logic)
- Verifica se um dos campos "sex" tem valor "checked", ou seja, se está selecionado.** (points to `d.sex[0].checked`)
- Verifica se uma das opções do campo "sex" foi selecionada.** (points to `!d.sex[0].checked && !d.sex[1].checked`)
- Se não foi encontrado nenhum problema nos campos, então esta função retorna verdadeiro, e assim o formulário é enviado para o servidor.** (points to `return true;`)

**Formulário Section:**

- Verifica se o campo "email" foi preenchido com um endereço de correio eletrônico.** (points to `d.email.value`)
- O atributo "onsubmit" recebe o retorno da função validaform. Enquanto este retorno for o valor falso os dados inseridos no formulário não são submetidos; e se o valor do retorno for verdadeiro, os dados são enviados para o programa indicado no atributo "action".** (points to `onsubmit="return validaForm()"`)



## Atividade 04

---

1. Escreva um documento HTML correspondente à aparência ilustrada na figura que segue. Observe que este exercício contém tabelas, listas e formulários.

**Figura 14** - Atividade - Formulário

Formulário de Anúncio d...

file:///C:/Users/Lirene/Desktop/FormAnuncioCurriculo.html

Galeria do Web Slice Sites Sugeridos Google Home Guia para o...

**Dados Pessoais**

\* Nome completo:

\* E-mail:

\* Data de nascimento:  ☐ Não exibir idade ao enviar o currículo às empresas.

\* Sexo: ☐ Masculino ☐ Feminino

\* Estado civil:

Homepage pessoal:

\* Está empregado atualmente? ☐ Não ☐ Sim  
Essa informação não será exibida em seu currículo.

**Área profissional desejada**

É necessária a escolha de pelo menos uma área profissional.

- ☐ Administração
- ☐ Educação
- ☐ Informática
- ☐ Hotelaria e Turismo
- ☐ Saúde
- ☐ Agricultura, Pecuária e Veterinária
- ☐ Artes, Arquitetura e Design
- ☐ Comunicação / Marketing
- ☐ Engenharia
- ☐ Jurídica
- ☐ Telemarketing

**Experiência profissional**

Recomendamos que você faça um resumo de sua experiência, conhecimentos e habilidades mais relevantes para o cargo pretendido.

\* Resumo do currículo:

Escreva no mínimo 1000 caracteres.

Os campos marcados com \* são de preenchimento obrigatório.

## Frames

---

Frames são divisões da tela do browser que tornam possível a apresentação de mais de uma página por vez, sendo cada uma independente da outra.

As desvantagens do uso dos frames são a dificuldade para imprimir uma página inteira e o fato de que o desenvolvedor web tem que manter mais de um documento simultaneamente. Além disso os frames não são mais suportados no HTML 5, apesar deles ainda funcionarem nos navegadores.

É importante aprender como funcionam os frames pois você pode acabar se deparando com um site que os utilize e precise realizar alguma mudança.

Para validar uma página com frames, é necessário selecionar o doctype apropriado:

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/fr
```

A inserção de frames é feita através das tags **<frameset> ...</frameset>**, em substituição ao elemento **<body>...</body>**.

O marcador **<frameset>** possui dois atributos, **rows** e **cols**. Esses atributos definem, respectivamente, o número de espaços horizontais (comprimento) e verticais (largura) do frame. Você não deve definir esses dois atributos ao mesmo tempo. Caso eles não sejam definidos, o valor default será acionado e o frame ocupará todo o espaço da página. Os valores podem ser dados em pixels ou em porcentagem. No exemplo a seguir, os atributos estão valorados com três dados, 20%, \* e 10%, devido à existência de três frames. A inserção de cada frame individual será vista a seguir.

```
1 <frameset rows="20%,*,10%" >...</frameset>
```

Cada frame (página) é inserido pela tag **<frame>**, que possui como atributos principais:

1) **src** – Especifica a página a ser exibida pelo frame. No exemplo a seguir, "frame1.html" é um documento HTML que deve conter as estruturas que você já aprendeu.

```
1 <frame src="frame1.html"/>
```

2) **name** – Especifica o nome do frame. Esse atributo tem grande importância pelo fato de toda navegação entre frames ser baseada no nome deles.

```
1 <frame src="frame1.html" name= "Frame 1"/>
```

3) **scrolling** – Determina a presença ou ausência de barras de rolagem junto àquela área do frameset. Os valores podem ser "auto", "yes" ou "no". O valor "auto" oferece barras de rolagem caso sejam necessárias e é o valor padrão. "yes" sempre oferece barras de rolagem para o frame mesmo que ele não precise, e "no" não oferece barras de rolagem para o frame mesmo que ele precise.

```
1 <frame src="frame1.html" scrolling="auto">
```

As tags <frame> e <frameset> não são mais suportadas no HTML5

A seguir, na **Figura 15**, observe um exemplo de código HTML, onde temos a inserção de 3 frames. Lembre-se de que a tag <frameset> substitui a tag <body>, ou seja, você não pode especificar um documento com ambas as seções. Os valores dos atributos rows e cols referem-se a cada frame inserido separadamente. O resultado desse conjunto de 3 documentos HTML é mostrado na **Figura 16**.

**Figura 15** - Exemplo de documento HTML com frames

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
  <head>
    <title>Clara Camarão</title>
  </head>
  <frameset rows="15%,*">
    <frame src="pag1.html" name="titulo"/>
    <frameset cols="20%,*">
      <frame src="pag2.html" name="menu"/>
      <frame src="pag3.html" name="principal"/>
    </frameset>
  </frameset>
</html>
```

**Figura 16** - Exemplo de página com frames



Com a divisão da tela em vários frames, é necessário indicar em qual dos frames um certo link deve ser aberto. Assim, ao definir links além do atributo **href**, é necessário especificar o atributo **target**, dando a ele o valor correspondente ao nome do frame. Assim, se quiséssemos que certo link abrisse no terceiro frame do documento anterior, teríamos que especificar o link da seguinte maneira:

```
1 <a href="pag3.html#A1" target="principal">Quem sou eu?</a>
2
3 Outros valores possíveis para target são "_blank" (para fazer com que esse link se abra em uma out
```

## O elemento <noframes>

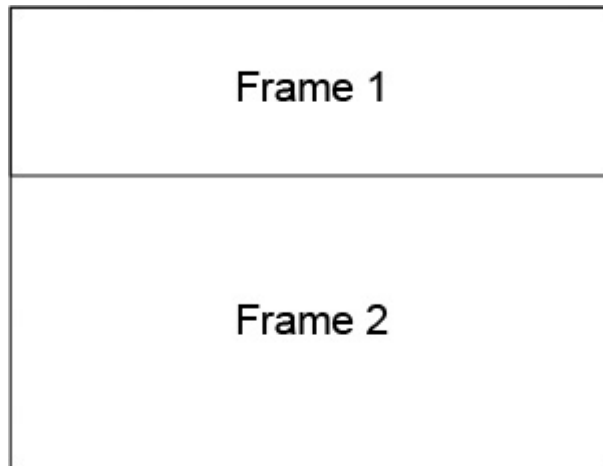
O elemento **<noframes>** é usado para indicar ao browser o que apresentar caso ele não dê suporte ao uso de frames. Assim, esse elemento serve para levar o usuário a uma versão sem frames daquele documento ou para exibir alguma mensagem. Essa mensagem pode conter quaisquer dos elementos utilizados dentro da tag **<body>**. A tag **<noframes>** deve ser escrita dentro do elemento **<frameset>**. Veja um exemplo a seguir.

```
1 <frameset>
2 ...
3 <noframes>
4 <body><p><a href="principal_semframes.html">Aqui você encontra uma versão sem frames desse
5 </noframes>
6 </frameset>
```

## Atividade 05

---

1. Reestruture o documento que você estava escrevendo sobre você em dois frames, o primeiro para acomodar o título e o menu e o segundo frame para acomodar as informações sobre você, de acordo com a figura que segue. Faça com que o frame 1 ocupe apenas o espaço suficiente para conter todo o menu e o segundo frame ocupe o restante do espaço.



## Conclusão

---

Bem, chegamos ao fim da nossa Aula 3. A partir de agora, você estudará as diferenças entre HTML e XHTML e então aprenderá a formatar documentos (X)HTML através de CSS, deixando-os muito mais interessantes e harmoniosos.

## Leitura complementar

---

- <http://www.w3.org/TR/1999/REC-html401-19991224/struct/objects.html>

Nesse site, você verá a diferença entre utilizar o atributo longdesc de <img/> e a tag <a> para criar um link em imagens.

## Resumo

---

Nesta aula, você aprendeu como definir figuras, tabelas, listas e frames em documentos HTML. Cada um desses recursos exige a memorização de muitas tags e atributos e, para isso, nada melhor que a prática.

## Autoavaliação

---

1. Escolha um dos temas a seguir e especifique documentos HTML (ligados entre si) que contenham todos os elementos descritos.

Temas: **esportes ou hobbies** (aquele com o qual você mais se identifica).

- a. Lista – deve conter pelo menos uma lista de cada tipo: ordenada, não ordenada e de descrição.
- b. Figura – deve conter pelo menos 2 figuras.
- c. Tabela – deve conter pelo menos uma tabela e ela deve ter rótulos nas colunas e legenda.
- d. Formulário – deve conter um formulário contendo os campos Nome, E-mail e Comentários, para que os usuários possam entrar em contato com você dando sugestões ou fazendo perguntas.

2. Faça uma versão desse site usando frames, como mostra a **Figura 16**.

**Observação:** Você deve guardar todos os arquivos referentes a esse exercício, pois eles serão usados nas próximas aulas.

## Referências

---

W3SCHOOL. Disponível em: <<http://www.w3schools.com/>>. Acesso em: 26 ago. 2015.

WIKIPÉDIA. **Applet**. Disponível em: <<http://pt.wikipedia.org/wiki/Applet>>. Acesso em: 26 ago. 2015.

W3c. Disponível em: <<http://www.w3.org/>>. Acesso em: 26 ago. 2015.